# Specification: Platform

## General

The Platform set of services contains services for monitoring and control of devices on-board of a spacecraft platform. The following services are contained in the Platform services set:

The Camera service allows a consumer to take and stream pictures.

The GNSS service allows a consumer to retrieve satellite navigation data.

The Autonomous ADCS service allows a consumer to determine the attitude, and additionally to control the attitude by selecting the desired definition.

The Software-defined Radio service allows a consumer to configure and receive a stream of data from a Software-defined Radio.

The Optical Data Receiver service allows a consumer to stream data from the Optical Data Receiver.

The Magnetometer service allows a consumer to acquire the magnetic field.

The Power Control service allows a consumer to control the power of the different subsystems.

This section details the Platform services. The area and structures are defined in terms of the MO Message Abstraction Layer (MAL), so it is possible to deploy them over any supported protocol and message transport.

## Service: Camera

### General

The Camera service allows a consumer to acquire pictures and control a camera in the spacecraft platform. The service can perform format conversions in case the consumer selects a specific format other than raw. The service can also stream pictures periodically.

Table 1‑1: Camera Service Operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Area Identifier | Service Identifier | Area Number | Service Number | Area Version |
| Platform | Camera | 105 | 1 | 1 |
| Interaction Pattern | Operation Identifier | Operation Number | Support in Replay | Capability Set |
| PUBLISH-SUBSCRIBE | [streamPictures](#_OPERATION_Camera_streamPictures) | 1 | No | 1 |
| SUBMIT | [setStreaming](#_OPERATION_Camera_setStreaming) | 2 | No |
| SUBMIT | [unsetStreaming](#_OPERATION_Camera_unsetStreaming) | 3 | No |
| REQUEST | [previewPicture](#_OPERATION_Camera_previewPicture) | 4 | No | 2 |
| INVOKE | [takePicture](#_OPERATION_Camera_takePicture) | 5 | No | 3 |
| REQUEST | [getProperties](#_OPERATION_Camera_getProperties) | 6 | Yes | 4 |

### High Level Requirements

1. The Camera service shall provide:
   1. the capability for streaming pictures;
   2. the capability for previewing pictures;
   3. the capability for taking pictures;
   4. the capability for requesting the service and camera properties.

### Functional Requirements

### OPERATION: streamPictures

#### General

The streamPictures operation allows a consumer to subscribe for picture updates.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | streamPictures | |
| Interaction Pattern | PUBLISH-SUBSCRIBE | |
| Pattern Sequence | Message | Body Type |
| OUT | PUBLISH/NOTIFY | picture : ([Picture](#_DATATYPE_Picture)) |

#### Structures

1. The picture in streaming mode.
2. The MAL EntityKey.firstSubKey shall be the value provided by the consumer during the setStreaming operation.
3. The MAL EntityKey.secondSubKey shall hold a unique object instance identifier.
4. The MAL EntityKey.thirdSubKey shall hold the width value of the resolution object provided by the consumer during the setStreaming operation.
5. The MAL EntityKey.fourthSubKey shall hold the height value of the resolution object provided by the consumer during the setStreaming operation.
6. The timestamp of the update shall be the time of when the picture was taken.

#### Errors

The operation does not return any errors.

### OPERATION: setStreaming

#### General

The setStreaming operation allows a consumer to set the rate, dimension and format of the Pictures being streamed.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | setStreaming | |
| Interaction Pattern | SUBMIT | |
| Pattern Sequence | Message | Body Type |
| IN | SUBMIT | streamingRate : (MAL::Duration)  resolution : ([PixelResolution](#_DATATYPE_PixelResolution))  format : ([PictureFormat](#_DATATYPE_PictureFormat))  firstEntityKey : (MAL::Identifier)  exposureTime : (MAL::Duration) |

#### Structures

1. The streamingRate field shall hold the streaming rate of the pictures.
2. The resolution field shall hold the desired streaming pictures resolution.
3. The format field shall hold the desired streaming picture format.
4. The firstEntityKey field holds the first entity key that will be present during the broadcast of the streamPictures operation.
5. The firstEntityKey field must not empty, nor be NULL nor the wildcard '\*'. An INVALID error shall be returned in this case.
6. The exposureTime field shall hold the exposure time of each frame.

#### Errors

The operation may return one of the following errors:

##### ERROR: INVALID

1. The requested streaming rate is not supported.
2. The extra information fied shall hold the minimum duration that can be selected.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | MAL::Duration |

##### ERROR: INVALID

1. The requested resolution is not supported.
2. The extra information fied shall hold the list of available resolutions.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | List<PixelResolution> |

##### ERROR: INVALID

1. The requested format is not supported.
2. The extra information fied shall hold the list of available formats.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | List<PictureFormat> |

##### ERROR: INVALID

The firstEntityKey field cannot be a null nor a wildcard.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | Not Used |

##### ERROR: DEVICE\_IN\_USE

The Camera is currently being used. Use the unsetStreaming operation in order to manually unset the streaming and try again.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_IN\_USE | 10509 | Not Used |

##### ERROR: DEVICE\_NOT\_AVAILABLE

The Camera unit is not available.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_NOT\_AVAILABLE | 10510 | Not Used |

### OPERATION: unsetStreaming

#### General

|  |  |  |
| --- | --- | --- |
| Operation Identifier | unsetStreaming | |
| Interaction Pattern | SUBMIT | |
| Pattern Sequence | Message | Body Type |
| IN | SUBMIT |  |

#### Structures

#### Errors

The operation does not return any errors.

### OPERATION: previewPicture

#### General

The previewPicture operation allows a consumer to quickly get a raw picture from the camera with a lower quality. This operation intendes to

provide a quick snap of what the camera is observing.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | previewPicture | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST |  |
| OUT | RESPONSE | picture : ([Picture](#_DATATYPE_Picture)) |

#### Structures

1. The picture field shall hold the picture preview.

#### Errors

The operation may return one of the following errors:

##### ERROR: DEVICE\_IN\_USE

The Camera is currently being used. Use the unsetStreaming operation in order to manually unset the streaming and try again.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_IN\_USE | 10509 | Not Used |

##### ERROR: DEVICE\_NOT\_AVAILABLE

The Camera unit is not available.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_NOT\_AVAILABLE | 10510 | Not Used |

### OPERATION: takePicture

#### General

The takePicture operation allows a consumer to take a picture from the camera. The appropriate dimension and format of the picture can be selected.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | takePicture | |
| Interaction Pattern | INVOKE | |
| Pattern Sequence | Message | Body Type |
| IN | INVOKE | resolution : ([PixelResolution](#_DATATYPE_PixelResolution))  format : ([PictureFormat](#_DATATYPE_PictureFormat))  exposureTime : (MAL::Duration) |
| OUT | ACK |  |
| OUT | RESPONSE | picture : ([Picture](#_DATATYPE_Picture)) |

#### Structures

1. The resolution field shall hold the desired picture resolution.
2. The format field shall hold the desired streaming picture format.
3. The exposureTime field shall hold the exposure time of the camera.
4. The picture field shall hold the picture.

#### Errors

The operation may return one of the following errors:

##### ERROR: INVALID

1. The requested resolution is not supported.
2. The extra information fied shall hold the list of available resolutions.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | List<PixelResolution> |

##### ERROR: INVALID

1. The requested format is not supported.
2. The extra information fied shall hold the list of available formats.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | List<PictureFormat> |

##### ERROR: DEVICE\_IN\_USE

The Camera is currently being used. Use the unsetStreaming operation in order to manually unset the streaming and try again.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_IN\_USE | 10509 | Not Used |

##### ERROR: DEVICE\_NOT\_AVAILABLE

The Camera unit is not available.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_NOT\_AVAILABLE | 10510 | Not Used |

### OPERATION: getProperties

#### General

The getProperties operation allows a consumer to request the properties of the service and camera. The service shall provide information about the available resolutions, formats and some extra information.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | getProperties | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST |  |
| OUT | RESPONSE | availableResolutions : (List<[PixelResolution](#_DATATYPE_PixelResolution)>)  availableFormats : (List<[PictureFormat](#_DATATYPE_PictureFormat)>)  extraInfo : (MAL::String) |

#### Structures

1. The availableResolutions field shall hold the list of the available resolutions by the camera.
2. The availableFormats field shall hold a list of available formats by the service.
3. The info field shall hold additional properties of the camera. The actual content is undefined.

#### Errors

The operation does not return any errors.

## Service: GPS

### General

The GPS service provides the ability to retrieve satellite navigation data from a Global Navigation Satellite System (GNSS) device receiver in the spacecraft platform. The GPS service provides the capability for streaming NMEA messages; the capability for enabling/disabling the streaming of NMEA messages; the capability for getting the last known position from the receiver; the capability for getting the satellites GNSS information; the capability for maintaining the list of nearby position events.

The nearbyPosition operation allows a consumer to receive a message from the service when the spacecraft enters or exists a certain position. These can be set using the addNearbyPosition and removed using the removeNearbyPosition.

Table 1‑1: GPS Service Operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Area Identifier | Service Identifier | Area Number | Service Number | Area Version |
| Platform | GPS | 105 | 2 | 1 |
| Interaction Pattern | Operation Identifier | Operation Number | Support in Replay | Capability Set |
| INVOKE | [getNMEASentence](#_OPERATION_GPS_getNMEASentence) | 1 | No | 1 |
| REQUEST | [getLastKnownPosition](#_OPERATION_GPS_getLastKnownPosition) | 2 | No | 2 |
| INVOKE | [getPosition](#_OPERATION_GPS_getPosition) | 3 | No | 3 |
| INVOKE | [getSatellitesInfo](#_OPERATION_GPS_getSatellitesInfo) | 4 | No | 4 |
| REQUEST | [listNearbyPosition](#_OPERATION_GPS_listNearbyPosition) | 5 | No | 5 |
| REQUEST | [addNearbyPosition](#_OPERATION_GPS_addNearbyPosition) | 6 | No | 6 |
| SUBMIT | [removeNearbyPosition](#_OPERATION_GPS_removeNearbyPosition) | 7 | No |
| PUBLISH-SUBSCRIBE | [nearbyPosition](#_OPERATION_GPS_nearbyPosition) | 8 | No | 7 |

### High Level Requirements

1. The GPS service shall provide:
   1. the capability for streaming GPS NMEA messages;
   2. the capability for enabling/disabling the streaming of NMEA messages;
   3. the capability for having getting the last known position from the GPS;
   4. the capability for getting the satellites GPS information;
   5. The capability for maintaining the list of nearby position events.

### Functional Requirements

### COM usage

Table 1‑1: GPS Service Object Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object Name | Object Number | Object Body Type | Related points to | Source points to |
| NearbyPosition | 1 | [NearbyPositionDefinition](#_DATATYPE_NearbyPositionDefinition) |  | The source link of the NearbyPosition object should be the object that caused it to be created, most likely a COM OperationActivity object or an operator login in the case of off-line editors being used. |
| NearbyPositionAlert | 2 | MAL::Boolean | 1 |  |

### COM Object Relationships

The Figure below shows the COM object relationships for this service:



Figure 1‑1: GPS Service COM object relationships

### COM Archive Service usage

1. NearbyPosition objects should be stored in the COM archive.
2. When a nearby event is published, the NearbyPosition object should be stored in the COM archive by the publisher.

### OPERATION: getNMEASentence

#### General

The getNMEASentence operation allows a consumer to request a NMEA sentence from a sentence identifier.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | getNMEASentence | |
| Interaction Pattern | INVOKE | |
| Pattern Sequence | Message | Body Type |
| IN | INVOKE | sentenceIdentifier : (MAL::String) |
| OUT | ACK |  |
| OUT | RESPONSE | sentence : (MAL::String) |

#### Structures

1. The sentenceIdentifier field shall hold the sentence identifier of the request.
2. The sentence field shall hold the NMEA sentence.

#### Errors

The operation may return one of the following errors:

##### ERROR: INVALID

The requested sentence identifier is invalid.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | Not Used |

##### ERROR: DEVICE\_NOT\_AVAILABLE

The GPS unit is not available.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_NOT\_AVAILABLE | 10510 | Not Used |

### OPERATION: getLastKnownPosition

#### General

The getLastKnownPosition operation allows a consumer to retrieve the last known position that was provided by the GPS unit.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | getLastKnownPosition | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST |  |
| OUT | RESPONSE | position : ([Position](#_DATATYPE_Position))  elapsedTime : (MAL::Duration) |

#### Structures

1. The position field shall hold the last known position.
2. The elapsedTime field shall hold the elapsed time since the position's determination.

#### Errors

The operation may return the following error:

##### ERROR: UNKNOWN

The position is unknown.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| UNKNOWN | Defined in MAL | Not Used |

### OPERATION: getPosition

#### General

The getPosition operation allows a consumer to obtain a position from the GPS unit.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | getPosition | |
| Interaction Pattern | INVOKE | |
| Pattern Sequence | Message | Body Type |
| IN | INVOKE |  |
| OUT | ACK |  |
| OUT | RESPONSE | position : ([Position](#_DATATYPE_Position)) |

#### Structures

1. The position field shall hold the current position.

#### Errors

The operation may return one of the following errors:

##### ERROR: UNKNOWN

The position is unknown.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| UNKNOWN | Defined in MAL | Not Used |

##### ERROR: DEVICE\_NOT\_AVAILABLE

The GPS unit is not available.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_NOT\_AVAILABLE | 10510 | Not Used |

### OPERATION: getSatellitesInfo

#### General

The getSatellitesInfo operation allows a consumer to obtain the satellites information from the GPS.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | getSatellitesInfo | |
| Interaction Pattern | INVOKE | |
| Pattern Sequence | Message | Body Type |
| IN | INVOKE |  |
| OUT | ACK |  |
| OUT | RESPONSE | gpsSatellitesInfo : (List<[SatelliteInfo](#_DATATYPE_SatelliteInfo)>) |

#### Structures

1. The gpsSatellitesInfo field shall hold the information of the satellites.

#### Errors

The operation may return the following error:

##### ERROR: DEVICE\_NOT\_AVAILABLE

The GPS unit is not available.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_NOT\_AVAILABLE | 10510 | Not Used |

### OPERATION: listNearbyPosition

#### General

The listNearbyPositionEvent operation allows a consumer to request the object instance identifiers of the existing nearby position events in the service. The operation is expected to be used in conjunction with the COM archive which holds the actual nearby position objects.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | listNearbyPosition | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST | names : (List<MAL::Identifier>) |
| OUT | RESPONSE | objInstIds : (List<MAL::Long>) |

#### Structures

1. The names field shall hold a list of NearbyPositionEvent names.
2. The objInstIds field shall hold a list of the corresponding object instance identifiers for the selected names.
3. The request may contain the wildcard value of '\*' to return all supported nearby position events.
4. The returned list shall maintain the same order as the submitted list unless the wildcard value was included in the request.

#### Errors

The operation does not return any errors.

### OPERATION: addNearbyPosition

#### General

The addNearbyPositionEvent operation allows a consumer to define a list of nearby position events in the service.The new NearbyPosition objects are expected to be stored in the COM Archive by the provider of the GPS service.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | addNearbyPosition | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST | nearbyPositionDefinitions : (List<[NearbyPositionDefinition](#_DATATYPE_NearbyPositionDefinition)>) |
| OUT | RESPONSE | objInstIds : (List<MAL::Long>) |

#### Structures

1. The nearbyPosition field shall hold the definition of a position.
2. The objInstIds field shall hold a list of the corresponding object instance identifiers for the selected names.

#### Errors

The operation may return one of the following errors:

##### ERROR: INVALID

1. One of the supplied attitudeDefinitions objects contains an invalid name.
2. The extra information field contains a list of the indexes of the erroneous values from the originating list supplied.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | List<MAL::UInteger> |

##### ERROR: DUPLICATE

1. One or more of the nearbyPosition objects being added has a name that is already in use in the domain.
2. The extra information field contains a list of the indexes of the erroneous values from the originating list supplied.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DUPLICATE | Defined in COM | MAL::UInteger |

### OPERATION: removeNearbyPosition

#### General

The removeNearbyPositionEvent operation allows a consumer to remove a list of nearby position events set in the service.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | removeNearbyPosition | |
| Interaction Pattern | SUBMIT | |
| Pattern Sequence | Message | Body Type |
| IN | SUBMIT | objInstIds : (List<MAL::Long>) |

#### Structures

1. The objInstIds field shall hold a list of the corresponding object instance identifiers of the NearbyPosition objects.

#### Errors

The operation may return the following error:

##### ERROR: UNKNOWN

1. One of the supplied NearbyPosition object instance identifiers is unknown.
2. A list of the indexes of the error values shall be contained in the extra information field.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| UNKNOWN | Defined in MAL | List<MAL::UInteger> |

### OPERATION: nearbyPosition

#### General

The nearbyPosition allows a consumer to subscribe to nearby position notifications. The notifications shall be generated upon entering or exiting a nearby position.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | nearbyPosition | |
| Interaction Pattern | PUBLISH-SUBSCRIBE | |
| Pattern Sequence | Message | Body Type |
| OUT | PUBLISH/NOTIFY | isEntering : (MAL::Boolean) |

#### Structures

1. The isEntering field shall hold the entering state of the current position.
2. If the GPS current position has entered the nearby position, then the field shall be set to TRUE otherwise FALSE.
3. The MAL EntityKey.firstSubKey shall contain the NearbyPosition name.
4. The MAL EntityKey.secondSubKey shall contain the NearbyPosition object instance identifier.
5. The MAL EntityKey.thirdSubKey shall contain the NearbyPositionAlert object instance identifier.
6. The MAL EntityKey.fourthSubKey shall be NULL.
7. The timestamp of the update shall be on-board clock at the time of the notification.
8. The ObjectId shall be set to NULL.

#### Errors

The operation does not return any errors.

## Service: AutonomousADCS

### General

The AutonomousADCS service allows a consumer to monitor the attitude from an ADCS device in the spacecraft platform and to set/unset the desired attitude from a list of attitude definitions.

Table 1‑1: AutonomousADCS Service Operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Area Identifier | Service Identifier | Area Number | Service Number | Area Version |
| Platform | AutonomousADCS | 105 | 3 | 1 |
| Interaction Pattern | Operation Identifier | Operation Number | Support in Replay | Capability Set |
| PUBLISH-SUBSCRIBE | [monitorAttitude](#_OPERATION_AutonomousADCS_monitorAttitude) | 1 | No | 1 |
| SUBMIT | [setDesiredAttitude](#_OPERATION_AutonomousADCS_setDesiredAttitude) | 2 | No | 2 |
| SUBMIT | [unsetAttitude](#_OPERATION_AutonomousADCS_unsetAttitude) | 3 | No |
| REQUEST | [listAttitudeDefinition](#_OPERATION_AutonomousADCS_listAttitudeDefinition) | 4 | No | 3 |
| REQUEST | [addAttitudeDefinition](#_OPERATION_AutonomousADCS_addAttitudeDefinition) | 5 | No | 4 |
| SUBMIT | [removeAttitudeDefinition](#_OPERATION_AutonomousADCS_removeAttitudeDefinition) | 6 | No |

### High Level Requirements

1. The Autonomous ADCS service shall provide:
   1. the capability for periodic monitoring of the spacecraft's attitude;
   2. the capability for setting and unsetting the spacecraft's desired attitude;
   3. the capability for maintaining the list of attitude definitions;
   4. the capability for listing the object instance identifiers for the attitude definitions.

### Functional Requirements

1. // Add the validation checks criteria

### COM usage

1. An AttitudeDefinitionBDot COM object represents the definition of a BDot attitude. The object body shall hold an AttitudeDefinitionBDot structure.
   1. The AttitudeDefinitionBDot COM object source link should be the object that caused it to be created, most likely a COM OperationActivity object.
2. An AttitudeDefinitionSingleSpinning COM object represents the definition of a Single Spinning attitude. The object body shall hold an AttitudeDefinitionSingleSpinning structure.
   1. The AttitudeDefinitionSingleSpinning COM object source link should be the object that caused it to be created, most likely a COM OperationActivity object.
3. An AttitudeDefinitionSunPointing COM object represents the definition of a Sun Pointing attitude. The object body shall hold an AttitudeDefinitionSunPointing structure.
   1. The AttitudeDefinitionSunPointing COM object source link should be the object that caused it to be created, most likely a COM OperationActivity object.
4. An AttitudeDefinitionTargetTracking COM object represents the definition of a Target Tracking attitude. The object body shall hold an AttitudeDefinitionTargetTracking structure.
   1. The AttitudeDefinitionTargetTracking COM object source link should be the object that caused it to be created, most likely a COM OperationActivity object.
5. An AttitudeDefinitionNadirPointing COM object represents the definition of a Nadir Pointing attitude. The object body shall hold an AttitudeDefinitionNadirPointing structure.
   1. The AttitudeDefinitionNadirPointing COM object source link should be the object that caused it to be created, most likely a COM OperationActivity object.

Table 1‑1: AutonomousADCS Service Object Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object Name | Object Number | Object Body Type | Related points to | Source points to |
| AttitudeDefinitionBDot | 2 | [AttitudeDefinitionBDot](#_DATATYPE_AttitudeDefinitionBDot) |  |  |
| AttitudeDefinitionSingleSpinning | 3 | [AttitudeDefinitionSingleSpinning](#_DATATYPE_AttitudeDefinitionSingleSpinning) |  |  |
| AttitudeDefinitionSunPointing | 4 | [AttitudeDefinitionSunPointing](#_DATATYPE_AttitudeDefinitionSunPointing) |  |  |
| AttitudeDefinitionTargetTracking | 5 | [AttitudeDefinitionTargetTracking](#_DATATYPE_AttitudeDefinitionTargetTracking) |  |  |
| AttitudeDefinitionNadirPointing | 6 | [AttitudeDefinitionNadirPointing](#_DATATYPE_AttitudeDefinitionNadirPointing) |  |  |

### COM Event Service usage

1. An ADCSProblemDetected COM event represents the detection of a problem on the ADCS. The object body can hold a blob of data containing implementation-specific data.
   1. The ADCSProblemDetected COM event shall be generated upon the detection of a problem with the ADCS.

Table 1‑1: AutonomousADCS Service Events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Event Name | Object Number | Object Body Type | Related points to | Source points to |
| ADCSProblemDetected | 1 | MAL::Blob |  |  |

### COM Object Relationships

The Figure below shows the COM object and event relationships for this service:

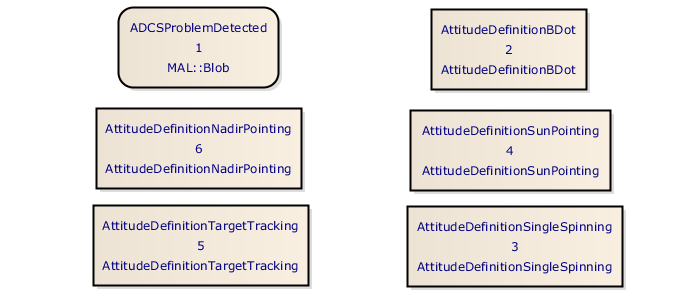


Figure 1‑1: AutonomousADCS Service COM object and event relationships

### COM Archive Service usage

1. AttitudeDefinitionBDot COM objects should be stored in the COM archive.
2. AttitudeDefinitionSingleSpinning COM objects should be stored in the COM archive.
3. AttitudeDefinitionSunPointing COM objects should be stored in the COM archive.
4. AttitudeDefinitionTargetTracking COM objects should be stored in the COM archive.
5. AttitudeDefinitionNadirPointing COM objects should be stored in the COM archive.

### OPERATION: monitorAttitude

#### General

The monitorAttitude operation allows a consumer to subscribe for the satellite's current attitude.

The pointingTo field shall contain the attitude instance in a reference frame.

The MAL EntityKey.firstSubKey shall contain the AttitudeDefinition name.

The MAL EntityKey.secondSubKey shall contain the AttitudeDefinition object instance identifier.

The MAL EntityKey.thirdSubKey shall contain the numeric value of the AttitudeMode of the AttitudeDefinition.

The MAL EntityKey.fourthSubKey shall be NULL.

The timestamp of the update shall be on-board clock at the time of the sampling.

The ObjectId shall be set to NULL.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | monitorAttitude | |
| Interaction Pattern | PUBLISH-SUBSCRIBE | |
| Pattern Sequence | Message | Body Type |
| OUT | PUBLISH/NOTIFY | attitudeInstance : ([AttitudeInstance](#_DATATYPE_AttitudeInstance)) |

#### Structures

1. The attitudeInstance filed shall hold the instance of the current attitude.

#### Errors

The operation does not return any errors.

### OPERATION: setDesiredAttitude

#### General

The setDesiredAttitude operation allows a consumer to set the spacecraft's attitude from an attitude definition.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | setDesiredAttitude | |
| Interaction Pattern | SUBMIT | |
| Pattern Sequence | Message | Body Type |
| IN | SUBMIT | objInstId : (MAL::Long)  autoUnset : (MAL::Duration)  streamingRate : (MAL::Duration) |

#### Structures

1. The object instance identifier of the attitude definition to be set.
2. The autoUnset field holds the duration after which the attitude definition shall be automatically unset.
3. If the field is null, then the attitude definition won't be automatically unset, in this case the manual unsetAttitude operation must be used to disengage it.
4. The streamingRate field shall contain the publishing frequency.

#### Errors

The operation may return one of the following errors:

##### ERROR: UNKNOWN

The submited object instance identifier is unknown.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| UNKNOWN | Defined in MAL | Not Used |

##### ERROR: INVALID

1. The selected attitude definition contains an invalid argument.
2. Contains the field name of the first field that did not pass the validation checks of the attitude definition.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | MAL::String |

##### ERROR: UNSUPPORTED\_OPERATION

The selected attitude definition is unsupported by the current implementation of the service.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| UNSUPPORTED\_OPERATION | Defined in MAL | Not Used |

##### ERROR: INVALID

1. The streamingRate is out of limits.
2. The minimum valid streamingRate.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | MAL::Duration |

##### ERROR: ADCS\_NOT\_AVAILABLE

The ADCS unit is not available.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| ADCS\_NOT\_AVAILABLE | 10 | Not Used |

##### ERROR: DEVICE\_IN\_USE

1. The ADCS is currently being used.
2. The service will unset the current selected attitude definition in the duration provided in the extra information field. If the extra information field is set to null, then the attitude definition shall not be automatically unset. The consumer should use the unsetAttitude operation to manually disengage it.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DEVICE\_IN\_USE | 10509 | MAL::Duration |

### OPERATION: unsetAttitude

#### General

The unsetAttitude operation allows a consumer to unset the spacecraft's attitude from the current selected attitude definition. This operation is expected to be used after a certain attitude definition is set by the setDesiredAttitude operation. If the attitude was set with the automatic unsetting enabled, then the unsetAttitude operation will override it and the attitude will be unset immediately.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | unsetAttitude | |
| Interaction Pattern | SUBMIT | |
| Pattern Sequence | Message | Body Type |
| IN | SUBMIT |  |

#### Structures

#### Errors

The operation may return the following error:

##### ERROR: ADCS\_NOT\_AVAILABLE

The ADCS unit is not available.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| ADCS\_NOT\_AVAILABLE | 10 | Not Used |

### OPERATION: listAttitudeDefinition

#### General

The listAttitudeDefinition operation allows a consumer to request the object instance identifiers of the attitude definition objects for the supported parameters of the provider. The operation is expected to be used in conjunction with the COM archive which holds the actual ParameterDefinition objects.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | listAttitudeDefinition | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST | names : (List<MAL::Identifier>) |
| OUT | RESPONSE | objIds : (List<MAL::Long>) |

#### Structures

1. The names of the Attitude definitions to be listed.
2. The names field may contain the wildcard value of '\*' to return all supported ParameterDefinition objects.
3. The objIds field holds the object instance identifiers of the Attitude definitions.
4. The returned list shall maintain the same order as the submitted list unless the wildcard value was included in the request.

#### Errors

The operation may return the following error:

##### ERROR: UNKNOWN

1. One of the supplied AttitudeDefinition object instance identifiers is unknown.
2. A list of the indexes of the error values shall be contained in the extra information field.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| UNKNOWN | Defined in MAL | List<MAL::UInteger> |

### OPERATION: addAttitudeDefinition

#### General

The addAttitudeDefinition operation allows a consumer to define one or more attitude definitions that do not currently exist. The new AttitudeDefinition object is expected to be stored in the COM archive by the provider of the AutonomousADCS service.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | addAttitudeDefinition | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST | attitudeDefinitions : (List<[AttitudeDefinition](#_DATATYPE_AttitudeDefinition)>) |
| OUT | RESPONSE | objIds : (List<MAL::Long>) |

#### Structures

1. The attitudeDefinitions field holds the attitude definitions.
2. The name field of the supplied AttitudeDefinition structures must not be NULL, the wildcard '\*', or empty. An INVALID error shall be returned in this case.
3. The supplied name must be unique among all AttitudeDefinition objects for the domain of the provider otherwise a DUPLICATE error shall be returned.
4. If an error is returned then no new attitude definitions shall be added as a result of this operation call.
5. The objIds field holds the object instance identifiers of the attitude definitions.
6. The returned list shall maintain the same order as the submitted definitions.

#### Errors

The operation may return one of the following errors:

##### ERROR: INVALID

1. One of the supplied attitudeDefinitions objects contains an invalid name.
2. The extra information field contains a list of the indexes of the erroneous values from the originating list supplied.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | List<MAL::UInteger> |

##### ERROR: DUPLICATE

1. One or more of the attitudeDefinitions objects being added has supplied a parameter name that is already in use in the domain.
2. The extra information field contains a list of the indexes of the erroneous values from the originating request list.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| DUPLICATE | Defined in COM | List<MAL::UInteger> |

##### ERROR: INVALID

1. The selected attitude definition contains an invalid argument.
2. Contains the field name of the first field that did not pass the validation checks of the attitude definition.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | MAL::String |

### OPERATION: removeAttitudeDefinition

#### General

The removeAttitudeDefinition operation allows a consumer to remove one or more attitude definitions from the list of attitudes supported by the service.

The operation does not remove the attitude definitions object from the COM archive, merely removes the object from the provider.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | removeAttitudeDefinition | |
| Interaction Pattern | SUBMIT | |
| Pattern Sequence | Message | Body Type |
| IN | SUBMIT | objIds : (List<MAL::Long>) |

#### Structures

1. The object instance identifiers of the attitude definition to be removed.
2. The list may contain the wildcard value of '0'.
3. If a provided ParameterDefinition object instance identifier does not include a wildcard and does not match an existing definition then this operation shall fail with an UNKNOWN error.
4. Matched attitude definition objects shall not be removed from the COM archive only from the list of attitude definitions objects in the provider.
5. If an error is raised then no definitions shall be removed as a result of this operation call.
6. If the operation succeeds then the provider shall no longer be able to set the removed attitude definitions.

#### Errors

The operation may return the following error:

##### ERROR: UNKNOWN

1. One of the supplied AttitudeDefinition object instance identifiers is unknown.
2. A list of the indexes of the error values shall be contained in the extra information field.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| UNKNOWN | Defined in MAL | List<MAL::UInteger> |

## Service: SoftwareDefinedRadio

### General

The Software-defined Radio provides a generic mechanism to set, configure and receive data from a Software-defined Radio device.

Table 1‑1: SoftwareDefinedRadio Service Operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Area Identifier | Service Identifier | Area Number | Service Number | Area Version |
| Platform | SoftwareDefinedRadio | 105 | 4 | 1 |
| Interaction Pattern | Operation Identifier | Operation Number | Support in Replay | Capability Set |
| PUBLISH-SUBSCRIBE | [streamRadio](#_OPERATION_SoftwareDefinedRadio_streamRadio) | 1 | No | 1 |
| SUBMIT | [enableSDR](#_OPERATION_SoftwareDefinedRadio_enableSDR) | 2 | No | 2 |
| SUBMIT | [updateConfiguration](#_OPERATION_SoftwareDefinedRadio_updateConfiguration) | 3 | No | 3 |

### High Level Requirements

### Functional Requirements

### OPERATION: streamRadio

#### General

The streamRadio operation allows a consumer to subscribe for Software-defined Radio data updates.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | streamRadio | |
| Interaction Pattern | PUBLISH-SUBSCRIBE | |
| Pattern Sequence | Message | Body Type |
| OUT | PUBLISH/NOTIFY | iqComponents : ([IQComponents](#_DATATYPE_IQComponents)) |

#### Structures

1. The iqComponents field shall hold the in-phase and quadrature data from the Software-defined Radio unit.

#### Errors

The operation does not return any errors.

### OPERATION: enableSDR

#### General

The enableSDR operation allows a consumer to enable or disable the Software-defined Radio.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | enableSDR | |
| Interaction Pattern | SUBMIT | |
| Pattern Sequence | Message | Body Type |
| IN | SUBMIT | enable : (MAL::Boolean)  initialConfiguration : ([SDRConfiguration](#_DATATYPE_SDRConfiguration))  publishingPeriod : (MAL::Duration) |

#### Structures

1. The enable field shall hold the flag status to enable/disable the Software-defined Radio.
2. If the enable field is true then the Software-defined Radio shall be enabled otherwise disabled.
3. The
4. initialConfiguration field shall hold the configuration to be set right after the SDR is enabled.

#### Errors

The operation may return one of the following errors:

##### ERROR: INVALID

The provided configuration is invalid.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | Not Used |

##### ERROR: INTERNAL

The Software-defined Radio could not be enabled.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INTERNAL | Defined in MAL | Not Used |

### OPERATION: updateConfiguration

#### General

The setConfiguration operation allows a consumer to set a configuration to the Software-defined Radio.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | updateConfiguration | |
| Interaction Pattern | SUBMIT | |
| Pattern Sequence | Message | Body Type |
| IN | SUBMIT | configuration : ([SDRConfiguration](#_DATATYPE_SDRConfiguration)) |

#### Structures

1. The configuration field holds the configuration of the SDR.

#### Errors

The operation may return the following error:

##### ERROR: INVALID

The provided configuration is invalid.

|  |  |  |
| --- | --- | --- |
| Error | Error # | ExtraInfo Type |
| INVALID | Defined in COM | Not Used |

## Service: OpticalDataReceiver

### General

The Optical Data Receiver service provides a mechanism to receive messages from an Optical Data Receiver device.

Table 1‑1: OpticalDataReceiver Service Operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Area Identifier | Service Identifier | Area Number | Service Number | Area Version |
| Platform | OpticalDataReceiver | 105 | 5 | 1 |
| Interaction Pattern | Operation Identifier | Operation Number | Support in Replay | Capability Set |
| PUBLISH-SUBSCRIBE | [streamData](#_OPERATION_OpticalDataReceiver_streamData) | 1 | No | 1 |
| REQUEST | [setPublishingFrequency](#_OPERATION_OpticalDataReceiver_setPublishingFrequency) | 2 | No |

### High Level Requirements

### Functional Requirements

### OPERATION: streamData

#### General

The streamData operation allows a consumer to subscribe to a stream feed from the Optical Data Receiver.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | streamData | |
| Interaction Pattern | PUBLISH-SUBSCRIBE | |
| Pattern Sequence | Message | Body Type |
| OUT | PUBLISH/NOTIFY | data : (MAL::Blob) |

#### Structures

1. The data field holds a blob of data coming from the Optical Data Receiver.

#### Errors

The operation does not return any errors.

### OPERATION: setPublishingFrequency

#### General

The setPublishingFrequency operation allows a consumer to set the publishing frequency or disable it.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | setPublishingFrequency | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST | publishingPeriod : (MAL::Duration) |
| OUT | RESPONSE |  |

#### Structures

1. The publishingPeriod field shall hold the period of updates to be streamed.
2. If set to 0, then no updates will be generated.

#### Errors

The operation does not return any errors.

## Service: Magnetometer

### General

The Magnetometer service provides a generic mechanism to retrieve the magnetic field from a magnetometer in the spacecraft platform.

Table 1‑1: Magnetometer Service Operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Area Identifier | Service Identifier | Area Number | Service Number | Area Version |
| Platform | Magnetometer | 105 | 6 | 1 |
| Interaction Pattern | Operation Identifier | Operation Number | Support in Replay | Capability Set |
| REQUEST | [getMagneticField](#_OPERATION_Magnetometer_getMagneticField) | 1 | No | 1 |

### OPERATION: getMagneticField

#### General

The getMagneticField operation allows a consumer to retrieve the magnetic field instance from a magnetometer unit.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | getMagneticField | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST |  |
| OUT | RESPONSE | magneticField : ([MagneticFieldInstance](#_DATATYPE_MagneticFieldInstance)) |

#### Structures

1. The magneticField field shall hold the magnetic field instance of the magnetometer.

#### Errors

The operation does not return any errors.

## Service: PowerControl

### General

The Power Control service provides a generic mechanism to list the available power units in a spacecraft platform and to enable/disable them.

Table 1‑1: PowerControl Service Operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Area Identifier | Service Identifier | Area Number | Service Number | Area Version |
| Platform | PowerControl | 105 | 7 | 1 |
| Interaction Pattern | Operation Identifier | Operation Number | Support in Replay | Capability Set |
| REQUEST | [listUnitsAvailable](#_OPERATION_PowerControl_listUnitsAvailable) | 1 | No | 1 |
| REQUEST | [enableUnit](#_OPERATION_PowerControl_enableUnit) | 2 | No | 2 |

### OPERATION: listUnitsAvailable

#### General

The listUnitsAvailable operation allows a consumer to request the list of the units available and their respective status.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | listUnitsAvailable | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST | names : (List<MAL::Identifier>) |
| OUT | RESPONSE | unitObjInstIds : (List<MAL::Long>)  enabled : (List<MAL::Boolean>) |

#### Structures

1. The names of the units.
2. The object instance identifier of the unit.
3. The status of the unit. True if enabled, false if disabled.

#### Errors

The operation does not return any errors.

### OPERATION: enableUnit

#### General

The enableUnit operation allows a consumer to enable and disable a specific Unit.

|  |  |  |
| --- | --- | --- |
| Operation Identifier | enableUnit | |
| Interaction Pattern | REQUEST | |
| Pattern Sequence | Message | Body Type |
| IN | REQUEST | enable : (MAL::Boolean)  unitObjInstIds : (MAL::Long) |
| OUT | RESPONSE |  |

#### Structures

1. Enable or disable the unit boolean.
2. The object instance identifier of the units to be enabled or disabled.

#### Errors

The operation does not return any errors.

# Data types

## Area data types: Platform

### Composite: Vector3D

The Vector3D holds a 3D Vector.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Vector3D | | |
| Extends | MAL::Composite | | |
| Short Form Part | 17 | | |
| Field | Type | Nullable | Comment |
| x | MAL::Double | Yes | The x component of the Vector. The unit shall be defined by each specific case. |
| y | MAL::Double | Yes | The y component of the Vector. The unit shall be defined by each specific case. |
| z | MAL::Double | Yes | The z component of the Vector. The unit shall be defined by each specific case. |

### Composite: WheelSpeed

The WheelSpeed holds the speed of the wheels.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | WheelSpeed | | |
| Extends | MAL::Composite | | |
| Short Form Part | 2 | | |
| Field | Type | Nullable | Comment |
| velocity | List<MAL::Double> | No | Velocity of the wheels in rpm. |

### Composite: Quaternions

The Quaternions holds the quaternion components.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Quaternions | | |
| Extends | MAL::Composite | | |
| Short Form Part | 16 | | |
| Field | Type | Nullable | Comment |
| q1 | MAL::Float | Yes | Quaternion component 1. Equivalent to the i component. |
| q2 | MAL::Float | Yes | Quaternion component 2. Equivalent to the j component. |
| q3 | MAL::Float | Yes | Quaternion component 3. Equivalent to the k component. |
| q4 | MAL::Float | Yes | Quaternion component 4. Equivalent to the l component. |

## Service data types: Camera

### Composite: Picture

The Picture structure holds the details of the picture.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Picture | | |
| Extends | MAL::Composite | | |
| Short Form Part | 1 | | |
| Field | Type | Nullable | Comment |
| content | MAL::Blob | No | The content of the picture. |
| creationDate | MAL::Time | Yes | The time of creation. |
| dimension | [PixelResolution](#_DATATYPE_PixelResolution) | Yes | The dimension of the picture. |
| format | [PictureFormat](#_DATATYPE_PictureFormat) | No | The format of the picture. |

### Composite: PixelResolution

The PixelResolution structure holds the pixel resolution of a picture.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PixelResolution | | |
| Extends | MAL::Composite | | |
| Short Form Part | 2 | | |
| Field | Type | Nullable | Comment |
| width | MAL::UInteger | No | The width of the picture. |
| height | MAL::UInteger | No | The height of the picture. |

### ENUMERATION: PictureFormat

PictureFormat is an enumeration definition holding the format of the picture.

|  |  |  |
| --- | --- | --- |
| Name | PictureFormat | |
| Short Form Part | 3 | |
| Enumeration Value | Numerical Value | Comment |
| RAW | 1 | RAW format. |
| BMP | 3 | Bitmap format. |
| PNG | 4 | PNG format. |
| JPG | 5 | JPG format. |

## Service data types: GPS

### Composite: Position

The Position structure holds the position of the spacecraft given by the GPS unit.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Position | | |
| Extends | MAL::Composite | | |
| Short Form Part | 1 | | |
| Field | Type | Nullable | Comment |
| latitude | MAL::Double | No | The latitude of the position. The unit is in decimal degrees. |
| longitude | MAL::Double | No | The longitude of the position. The unit is in decimal degrees. |
| altitude | MAL::Double | No | Altitude in meters according to the WGS-84 ellipsoid |
| extraDetails | [PositionExtraDetails](#_DATATYPE_PositionExtraDetails) | Yes | Extra details about the position. |

### Composite: SatelliteInfo

The SatelliteInfo structure holds the information about the satellites obtained from the GPS unit.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | SatelliteInfo | | |
| Extends | MAL::Composite | | |
| Short Form Part | 2 | | |
| Field | Type | Nullable | Comment |
| azimuth | MAL::Float | Yes | The azimuth of the satellite. |
| elevation | MAL::Float | Yes | The elevation of the satellite. |
| prn | MAL::Integer | Yes | The pseudorandom noise number. |
| almanac | MAL::Double | Yes | The almanac of the satellite. |
| ephemeris | MAL::Double | Yes | The ephemeris of the satellite. |
| recentFix | MAL::Time | Yes | The time of the fix. |
| svn | MAL::UInteger | Yes | Space vehicle number. |

### Composite: PositionExtraDetails

The PositionExtraDetails structure holds extra information that can be provided by a GPS unit.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PositionExtraDetails | | |
| Extends | MAL::Composite | | |
| Short Form Part | 4 | | |
| Field | Type | Nullable | Comment |
| utc | MAL::Time | Yes | UTC time status of position. |
| fixQuality | MAL::Integer | Yes | Fix Quality: 0 = Invalid; 1 = GPS fix; 2 = DGPS fix. |
| numberOfSatellites | MAL::Integer | Yes | The number of satellites in view for this position. |
| hdop | MAL::Float | Yes | Horizontal Dilution of Precision. The relative accuracy of horizontal position. |
| undulation | MAL::Float | Yes | The relationship between the geoid and the WGS84 ellipsoid. The unit is meters. |

### Composite: NearbyPositionDefinition

The NearbyPositionDefinition structure holds a definition of a certain Position.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | NearbyPositionDefinition | | |
| Extends | MAL::Composite | | |
| Short Form Part | 26 | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | Yes | The name of the definition. |
| description | MAL::String | Yes | A textual description of the definition. |
| distanceBoundary | MAL::Double | Yes | The distance boundary of this definition. The distance boundary shall define a spherical zone around the position. The unit is meters. |
| position | [Position](#_DATATYPE_Position) | Yes | The position of the definition. |

## Service data types: AutonomousADCS

### ENUMERATION: AttitudeMode

AttitudeMode is an enumeration definition holding the attitude modes.

|  |  |  |
| --- | --- | --- |
| Name | AttitudeMode | |
| Short Form Part | 4 | |
| Enumeration Value | Numerical Value | Comment |
| BDOT | 1 | B dot mode. Can also be used for detumbling and momentum unloading. |
| SUNPOINTING | 2 | Sun Pointing mode. |
| SINGLESPINNING | 3 | Single Spinning mode. |
| TARGETTRACKING | 4 | Target Pointing mode. |
| NADIRPOINTING | 5 | Nadir Pointing mode. |

### ENUMERATION: ReferenceFrame

ReferenceFrame is an enumeration definition holding the frames of reference.

|  |  |  |
| --- | --- | --- |
| Name | ReferenceFrame | |
| Short Form Part | 5 | |
| Enumeration Value | Numerical Value | Comment |
| EULER\_ANGLES | 1 | Euler angles reference frame. |
| UNIT\_QUATERNIONS | 2 | Quaternions reference frame. |
| ROTATION\_MATRIX | 3 | Rotation Matrix reference frame. |

### Composite: AttitudeInstance

The AttitudeInstance holds an attitude instance.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeInstance | | |
| Extends | MAL::Composite | | |
| Abstract | | | |

### Composite: AttitudeInstanceBDot

The AttitudeInstanceBDot structure holds an attitude instance of type BDot.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeInstanceBDot | | |
| Extends | [AttitudeInstance](#_DATATYPE_AttitudeInstance) | | |
| Short Form Part | 11 | | |
| Field | Type | Nullable | Comment |
| magneticField | [Vector3D](#_DATATYPE_Vector3D) | No | The magnetic field instance. |
| wheelSpeed | [WheelSpeed](#_DATATYPE_WheelSpeed) | Yes | The wheels speed instance. |
| mtqDipoleMomentum | [Vector3D](#_DATATYPE_Vector3D) | Yes | The magnetorquer momentum. |

### Composite: AttitudeInstanceSunPointing

The AttitudeInstanceSunPointing structure holds an attitude instance of type Sun Pointing.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeInstanceSunPointing | | |
| Extends | [AttitudeInstance](#_DATATYPE_AttitudeInstance) | | |
| Short Form Part | 12 | | |
| Field | Type | Nullable | Comment |
| sunVector | [Vector3D](#_DATATYPE_Vector3D) | No | The sun vector instance. |
| valid | MAL::Boolean | No | The validity of the instance. |
| wheelSpeed | [WheelSpeed](#_DATATYPE_WheelSpeed) | Yes | The wheels speed instance. |
| mtqDipoleMomentum | [Vector3D](#_DATATYPE_Vector3D) | Yes | This value shall not be used |

### Composite: AttitudeInstanceSingleSpinning

The AttitudeInstanceSingleSpinning structure holds an attitude instance of type Single Spinning.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeInstanceSingleSpinning | | |
| Extends | [AttitudeInstance](#_DATATYPE_AttitudeInstance) | | |
| Short Form Part | 13 | | |
| Field | Type | Nullable | Comment |
| sunVector | [Vector3D](#_DATATYPE_Vector3D) | No | The sun vector instance. |
| magneticField | [Vector3D](#_DATATYPE_Vector3D) | No | The magnetic field instance. |
| currentQuaternions | [Quaternions](#_DATATYPE_Quaternions) | No | The quaternions of the current attitude instance. |
| angularMomentum | [Vector3D](#_DATATYPE_Vector3D) | Yes | The angular momentum vector. |
| mtqDipoleMomentum | [Vector3D](#_DATATYPE_Vector3D) | Yes | The magnetorquer momentum. |

### Composite: AttitudeInstanceTargetTracking

The AttitudeInstanceTargetTracking structure holds an attitude instance of type Target Tracking.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeInstanceTargetTracking | | |
| Extends | [AttitudeInstance](#_DATATYPE_AttitudeInstance) | | |
| Short Form Part | 14 | | |
| Field | Type | Nullable | Comment |
| currentQuaternions | [Quaternions](#_DATATYPE_Quaternions) | No | The quaternions of the current attitude instance. |
| targetQuaternions | [Quaternions](#_DATATYPE_Quaternions) | No | The quaternions of the target attitude instance. |
| wheelSpeed | [WheelSpeed](#_DATATYPE_WheelSpeed) | Yes | The wheels speed instance. |
| positionVector | [Vector3D](#_DATATYPE_Vector3D) | Yes | WGS84 reference. |
| angularVelocity | [Vector3D](#_DATATYPE_Vector3D) | Yes | The angular velocity in rpm. |

### Composite: AttitudeInstanceNadirPointing

The AttitudeInstanceNadirPointing structure holds an attitude instance of type Nadir Pointing.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeInstanceNadirPointing | | |
| Extends | [AttitudeInstance](#_DATATYPE_AttitudeInstance) | | |
| Short Form Part | 15 | | |
| Field | Type | Nullable | Comment |
| currentQuaternions | [Quaternions](#_DATATYPE_Quaternions) | No | The quaternions of the current attitude instance. |
| targetQuaternions | [Quaternions](#_DATATYPE_Quaternions) | No | The quaternions of the target attitude instance. |
| speed | [WheelSpeed](#_DATATYPE_WheelSpeed) | Yes | The wheels speed instance. |
| positionVector | [Vector3D](#_DATATYPE_Vector3D) | Yes | WGS84 reference. |
| angularVelocity | [Vector3D](#_DATATYPE_Vector3D) | Yes | The angular velocity in rpm. |

### Composite: AttitudeDefinition

The AttitudeInstance holds an attitude definition.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeDefinition | | |
| Extends | MAL::Composite | | |
| Abstract | | | |
| Field | Type | Nullable | Comment |
| name | MAL::Identifier | Yes | The name of the attitude definition. |
| description | MAL::String | Yes | The description of the attitude definition. |

### Composite: AttitudeDefinitionBDot

The AttitudeDefinitionBDot structure holds an attitude definition of type BDot.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeDefinitionBDot | | |
| Extends | [AttitudeDefinition](#_DATATYPE_AttitudeDefinition) | | |
| Short Form Part | 21 | | |

### Composite: AttitudeDefinitionSingleSpinning

The AttitudeDefinitionSingleSpinning structure holds an attitude definition of type Sun Pointing.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeDefinitionSingleSpinning | | |
| Extends | [AttitudeDefinition](#_DATATYPE_AttitudeDefinition) | | |
| Short Form Part | 22 | | |
| Field | Type | Nullable | Comment |
| bodyAxis | [Vector3D](#_DATATYPE_Vector3D) | No | The axis defined in the body reference frame for spinning. |
| angularVelocity | MAL::Double | No | The magnitude of the angular velocity of the spinning in rpm. The right-hand rule shall be followed to indicate the positive direction of the angular velocity. |

### Composite: AttitudeDefinitionSunPointing

The AttitudeDefinitionSunPointing structure holds an attitude definition of type Single Spinning.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeDefinitionSunPointing | | |
| Extends | [AttitudeDefinition](#_DATATYPE_AttitudeDefinition) | | |
| Short Form Part | 23 | | |

### Composite: AttitudeDefinitionTargetTracking

The AttitudeDefinitionTargetTracking structure holds an attitude definition of type Target Tracking.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeDefinitionTargetTracking | | |
| Extends | [AttitudeDefinition](#_DATATYPE_AttitudeDefinition) | | |
| Short Form Part | 24 | | |
| Field | Type | Nullable | Comment |
| latitude | MAL::Double | No | The latitude of the target to be tracked. The unit is in decimal degrees. |
| longitude | MAL::Double | No | The longitude of the target to be tracked. The unit is in decimal degrees. |

### Composite: AttitudeDefinitionNadirPointing

The AttitudeDefinitionNadirPointing structure holds an attitude definition of type Nadir Pointing.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | AttitudeDefinitionNadirPointing | | |
| Extends | [AttitudeDefinition](#_DATATYPE_AttitudeDefinition) | | |
| Short Form Part | 25 | | |

## Service data types: SoftwareDefinedRadio

### Composite: SDRConfiguration

It holds a configuration for the SDR.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | SDRConfiguration | | |
| Extends | MAL::Composite | | |
| Short Form Part | 1 | | |
| Field | Type | Nullable | Comment |
| LOFrequency | MAL::Double | Yes | Local Oscillator Frequency. |
| LNA | MAL::Integer | Yes | Low-noise Amplifier Gain. |
| VGA1 | MAL::Integer | Yes | Variable Gain Amplifier 1. |
| VGA2 | MAL::Integer | Yes | Variable Gain Amplifier 2. |
| LPF | MAL::Double | Yes | Low Pass Filter frequency in GHz. |
| sampleRate | MAL::Double | Yes | The frequency in MHz. |

### Composite: IQComponents

It holds the In-phase and Quadrature components.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | IQComponents | | |
| Extends | MAL::Composite | | |
| Short Form Part | 27 | | |
| Field | Type | Nullable | Comment |
| inPhase | List<MAL::Double> | Yes | The in-phase component. |
| quadrature | List<MAL::Double> | Yes | The quadrature component. |

## Service data types: Magnetometer

### Composite: MagneticFieldInstance

An instance of the magnetic field vector.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | MagneticFieldInstance | | |
| Extends | MAL::Composite | | |
| Short Form Part | 1 | | |
| Field | Type | Nullable | Comment |
| x | MAL::Double | No | The X component of the magetic field in microTesla. |
| y | MAL::Double | No | The Y component of the magetic field in microTesla. |
| z | MAL::Double | No | The Z component of the magetic field in microTesla. |

# Error codes

The following table lists the errors defined in this specification:

Table 1‑1: Platform Error Codes

|  |  |  |
| --- | --- | --- |
| Error | Error # | Comment |
| DEVICE\_IN\_USE | 10509 | The device is currently in use. |
| DEVICE\_NOT\_AVAILABLE | 10510 | The device is currently not available. |
| ADCS\_NOT\_AVAILABLE | 10 | The ADCS is not available. |