Application Layer software may be DO-178C DAL E but will be assessed on a case by case basis.

ID : SSS_877

1AWObjectType : Requirement

The Platform Software shall be developed in accordance with RTCA/DO-178C, ED-12B Design Assurance Level (DAL) D.

ID : SSS_878

1AWObjectType : Requirement

The Platform Software shall be developed in accordance with RTCA/DO-330, DO-331, DO-332, and DO-333, as applicable.

ID : SSS_875

1AWObjectType : Title

3.3.5.2 Hardware Design Assurance Level

ID : SSS_879

1AWObjectType : Requirement

The hardware in the ADG-400 shall be developed to Level D in accordance with the criticality definition provided in RTCA/DO 254, ED-80.

ID : SSS_24

1AWObjectType : Title

3.3.6.1 ADG-400 Configuration Management

ID : SSS_127

1AWObjectType : Description

This section defines the structure and syntax of the configuration files which are loaded into the Honeywell Aircraft Data Gateway-400 unit (ADG-400). Configuration files allow customization of the ADG-400 operation for a specific airline/fleet. This section also describes the method of signing and encrypting configuration files, and the method of loading configuration files into the ADG-400.

ID : SSS_128

1AWObjectType : Title

3.3.6.1.1 ADG-400 Configuration Files

ID : SSS_129

1AWObjectType : Description

The ADG-400 contains a set of configuration files that allow the ADG-400 to be customized for a particular aircraft/fleet. A set of three configuration files is composed of a Honeywell Configuration File (HCF), an Aircraft Configuration File (ACF), and a User Configuration File (UCF). All of these files are individually loadable on the aircraft. The HCF is considered a Parameter Data Item (PDI) item per DO-178C and thus its configuration is controlled and maintained by Honeywell. The ACF and UCF are considered user-modifiable software items and therefore do not require FAA certification oversight. The UCF is maintained by airlines or aircraft manufacturers.

ID : SSS_2330

1AWObjectType : Requirement

The ADG-400 **shall** be compliant with Honeywell ADG-400 Configuration Tool - ICD TN-9040xxxx.

Note: The ADG-400 Configuration Tool - ICD ICD TN-9040xxxx defines the structure and syntax of the configuration files which are loaded into the Honeywell One Aero Wireless Unit (ADG-400). Configuration files allow customization of the ADG-400's operation for a specific airline/fleet. This document also describes the method of signing and encrypting configuration files, and the method of loading configuration files into the ADG-400.

ID : SSS_157

1AWObjectType : Requirement

The ADG-400 **shall** support configurable parameters.

ID : SSS_158

1AWObjectType : Requirement

The ADG-400 configuration file set **shall** contain, at a minimum, three elements:

- A User Configuration File (UCF)
- A Honeywell Configuration File (HCF)
- An Aircraft Configuration File (ACF)

ID : SSS_159

1AWObjectType : Requirement

The UCF and ACF **shall** be considered User Modifiable Software.

ID : SSS_160

1AWObjectType : Requirement

The HCF **shall** be considered a Parameter Data Item File.

ID : SSS_161

1AWObjectType : Description

The context of User Modifiable Software and Parameter Data Item is DO-178C.

ID : SSS_2386

1AWObjectType : Description

The ADG-400 Configuration Tool will generate signed and encrypted secure HCF/ACF/UCF files. These files includes text file formatted in JSON.

ID : SSS_2387

1AWObjectType : Description

Refer below for the recommended syntax and data type in JSON format.

In JSON,

- JSON objects are surrounded by curly braces {}.
- JSON objects are written in key/value pairs.
- A key/value pair consists of a field name (in double quotes), followed by a colon, followed by a value.
- Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null).
- Keys and values are separated by a colon.
- Each key/value pair is separated by a comma.
- Square brackets hold arrays.
- Array values must be of type string, number, object, array, boolean or null.
- Arrays in JSON Objects can be values of an object property.

In JSON, values must be one of the following data types:

- a string
- a number
- an object (JSON object)
- an array

- a boolean
- null

JSON values cannot be one of the following data types:

- a function
- a date
- undefined

Example

{

"name":"John",

"age":30,

"cars":["Ford", "BMW", "Fiat"]

}

ID : SSS_130

1AWObjectType : Title

3.3.6.1.1.1 Configuration File Contents

ID : SSS_4701

1AWObjectType : Description

This section contains the requirement of each of the configuration files available of ADG-400 system.

Note:

1. Functional Area coloumn added in each of the configuration table describes the ADG-400 functional area which may use the configuration parameter for controlling the system and setting up the user preferences.

2. Default coloumn added in each of the configuration table includes the default value which ADG-400 system uses for configuring the system when configuration data is not available or corrupted in SSD memory.

ID : SSS_4716

1AWObjectType : Description

Refer below for the list of terminology with the details used for defining the character maximum length, range, various data type etc. used in each of the table defining the configuration file format.

- TEXT means any valid ASCII printable character (32-126).
- IPADDRESS means a dotted decimal IP Address (X.Y.Z.A)
- SIGNED means a signed decimal non fractional number.
- UNSIGNED means a unsigned decimal non fractional number.
- HEX means raw hex (not 0x denoted).
- FRACTIONAL means a signed fractional decimal number.
- OCTAL means a octal number (not 0 front denoted).
- COUNTRYCODE means a valid country code (three-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard).
- AIRPORTCODE means a valid ICAO aiport code.
- WIFICHANNEL means a non overlapping WLAN channel defined under 802.11 b/g/n/ac.
- ALLOWED_VALUES refer to the allowed value defined in the description.
- SIZE means a unsigned decimal non fractional number defining the array size.
- RANGE means any value in between the minumum value and the maximum value.
- IPMASK means a dotted decimal IPV4 subnet address (A.B.C.D)
- PARTITION_NAME is the partition module to which the storage space is allocated. Refer SSS_4711 for the partition module and the space allocated to each module.

ID : SSS_131

1AWObjectType : Title

3.3.6.1.1.1.1 HCF Contents

ID : SSS_134

1AWObjectType : Description

The HCF contains elements related to FAA certification/regulatory requirements, overall security, radio parameters and ADG-400 network configuration parameters. Thus, the same HCF may be used for an entire aircraft fleet. The HCF file is created by Honeywell and provided to aircraft manufactures and airlines via the Honeywell SIGNS web portal. The HCF is also encrypted and digitally signed for security purposes when received by aircraft manufactures and airlines from the Honeywell.

ID : SSS_135

1AWObjectType : Description

The HCF will be composed of a single archive file named:

- key_enc_sign_hcf_N.zip if current certificates and keys are used, where “N” is a number that can be used for identification purposes (such as a software part number or version number).

- key_enc_sign_hcf_000000_factory.zip if factory default certificates are used.

In addition, both key_enc_sign_hcf_N.zip and key_enc_sign_hcf_000000_factory.zip files may be placed in a single zip file named gs_key_enc_sign_hcf_60004873_N.zip and uploaded to the ADG-400.

ID : SSS_136

1AWObjectType : Description

Each text file in the HCF file is formatted in JSON format.

ID : SSS_4179

1AWObjectType : Requirement

The enc_sign_hcf_N.zip archive file in the ADG-400 **shall** contain the following files:

- HCFheader.txt
- firewall.txt
- shorewall.zip
- whiteList.txt
- authorizationList.txt
- lans.txt
- hardwaremode.txt
- softwareconfig.txt
- config/HCFGroundEncryptKey.private
- config/HCFGroundEncryptKey.passphrase
- config/HCFSigningCA.CER
- config/HCFSigningCA.CRL

- radiusserver/radiusServerKey.p12
- radiusserver/wifiClientCA.CER
- radiusserver/wifiClientCA.CRL
- sslserver/standardServerSSLKey.p12
- sslserver/standardClientSSLCA.CER
- sslserver/standardClientSSLCA.CRL

ID : SSS_4180

1AWObjectType : Requirement

The HCFheader.txt file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
part_number	String	15, TEXT	Part Number associated with the HCF container	Field Not Specified	Configuration Manager
version	String	6, UNSIGNED	Version number of the HCF format	"000000"	Configuration Manager
creation_date	String	19, TEXT	creation date and time in MM/DD/YYYY HH:MM:SS format	"01/01/2016 12:00:00"	Configuration Manager
digest	String	64, TEXT	SHA-256 digest of all of the files in the HCF excluding the header.txt in alphabetical order	Field Not Specified	Configuration Manager

Example HCFheader.txt:

```
{
        "part_number": "60004872",
        "version": "000001",
        "creation_date": "01/15/2017 09:15:01",
        "digest": "5e7fb324d9c54ee0f61a3123648e4ff9724d452bba89edde884bb67a15e65399" ,
        }
```

ID : SSS_4181

1AWObjectType : Requirement

The firewall.txt file **shall** be formatted in JSON format and contains the following fields

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
firewall_script	String	64, TEXT	Name of the firewall script	Field Not Specified	Network Manager, Security

Example firewall.txt file

{

"firewall_script":"shorewall.zip"

}

ID : SSS_4182

1AWObjectType : Description

The components of the shorwall.zip and a brief description of the components are as follows. See http://shorewall.net for details and description of the firewall configuration.

Zones - Defines three zones. The following zones are defined:

loc (aircraft LAN)

rem (Internet)

fw (the firewall itself or ADG-400)

Interfaces - Defines the interfaces and the zone they belong to. The following interfaces are defined:

wlan0 for WiFi

usb0 for cellular

eth0 for LAN

Policy - Defines the generic behavior for traffic between zones. The following behavior is defined:

loc to fw is all allowed (aircraft LAN to ADG-400)

fw to all is allowed (ADG-400 to anywhere)

rem to all is dropped (anything from Internet is dropped unless a rule allows it)

Rules - Defines the exceptions to the policy file. The following exceptions are defined:

FTP connections from WiFi only allowed in AP Mode

Services like RADIUS, DNS, SSH and ICMP (ping) allowed for different interfaces

Masq - Generic masquerading definition for private networks

ID : SSS_4183

1AWObjectType : Requirement

The authorizationList.txt file **shall** be formatted in JSON format and contains an array of the following fields

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
area	String	ALLOWED_VALUES	Application which accepts this authorization. Allowed values are: (Default, WDLS, WDLS-local, LOGS, CONFIG, DATA, FPLANS, maintenance, LoadingSoftware, CRL)	Field Not Specified	Data Transfer
authorization_key	Array of String	256, TEXT	Base64 encoded Username:Password for the service. More than one record may exist for the same application.	Field Not Specified	Data Transfer
security	String	ALLOWED_VALUES	Allowed values are signingNencryption, signing, encryption, none, verification, decryption, decryptionNverification, uncrating, uncratingWithCRL, uncratingWithTS.	Field Not Specified	Data Transfer
area_type	String	ALLOWED_VALUES	Allowed values are Inbound (or) outbound (or) local	Field Not Specified	Data Transfer
policy	string	ALLOWED_VALUES	Allowed values are Unzip, move (or) none. When the policy is 'unzip', extracted content should move to "area-unzip" folder.	Field Not Specified	Data Transfer
policyArea	String	64, TEXT	To which the files should be moved when policy type is move.	Field Not Specified	Data Transfer
signingCa	String (optional)	64, TEXT	Used to validate public signing certificate for inbound operations.	Field Not Specified	Data Transfer
signingCrl	String (optional)	64, TEXT	Used to validate public signing certificate for inbound operations.	Field Not Specified	Data Transfer
signingPrivateKey	String (optional)	64, TEXT	Used to create signature file for outbound operations.	Field Not Specified	Data Transfer
signingPrivateKeyPassphrase	String(optional)	64, TEXT	Used to create signature file for outbound operations.	Field Not Specified	Data Transfer
encryptionPublicCertificate	String (optional)	64, TEXT	Used to encrypt random generated symmetric key which interns used to encrypt the data for outbound operations.	Field Not Specified	Data Transfer
encryptionPrivateKey	String (optional)	64, TEXT	Used to decrypt random generated symmetric key which intern used to decrypt the data for inbound operations.	Field Not Specified	Data Transfer
encryptionPrivateKeyPassphrase	String (optional)	256, TEXT	Passphrase for encryption private key for inbound operations.	Field Not Specified	Data Transfer
verificationCa	String (optional)	64, TEXT	Used to validate the crate when security set to uncratingWithCRL for inbound operation.	Field Not Specified	Data Transfer
verificationCrl	String (optional)	64, TEXT	Used to validate the crate when security set to uncratingWithCRL for inbound operations.	Field Not Specified	Data Transfer
tsaCa	String (optional)	64, TEXT	Used to validate the time Stamping Response when security set to uncrateWithTS for inbound operations.	Field Not Specified	Data Transfer

Note: Below are area description what kind of files will be available.

- WDLS: Contain Loadable LSAP files
- LOGS: ADG-400 log files like security and activity log
- CONFIG: ADG-400 configuration files HCF, UCF
- DATA: This folder contain data like QAR, Ethernet, etc.

- FPLANS: This area will contain flight plans
- MAINTENANCE: This area will contain downloaded maintenance data.
- LOADINGSOFTWARE: ADG-400 self-loading software.
- Default: For authorizing the common client operations

ID : SSS_4186

1AWObjectType : Description

Refer below for the example of authorizationList.txt:

```
[
 {
  "area": "WDLS",
  "authorization_key": [
   "ashkhgkagiuybfb132==",
   "jndr873rdcckn&e4f9nnr"
  ],
  “security”:”verification”,
  “area_type”:”inbound”,
  “policy”:”move”,
  “poicyArea”:”Avionics”,
  “signingCa”:”wdls/signingCA.CER”,
  “signingCrl”:”wdls/signingCA.CRL”,
 },
 {
  "area": "LOGS",
  "authorization_key": "ashkhgkagiuybfb132==",
  “area_type”:”outbound”,
  “policy”:”none”,
  “security”:”none”
 },
{
```

```
   "area": "LoadingSoftware",
   "authorization_key": "ashkhgkagiuybfb132==",
   “security”:”verification”,
   “area_type”:”inbound”,
   “policy”:”none”,
   “signingCa”:”loadingSoftware/signingCA.CER”,
   “signingCrl”:” loadingSoftware/signingCA.CRL”,
  },
{
   "area": "Data",
   "authorization_key": "ashkhgkagiuybfb132==",
   “security”:”encryption”,
   “area_type”:”outbound”
   “encryptionPublicCertificate”:”data/encryptionPublic.CER”
  },

]
```

ID : SSS_4771

1AWObjectType : Requirement

The whiteList.txt file **shall** be formatted in JSON format and contains the following fields

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
description	String (Optional)	64, TEXT	Name of the firewall script	Field Not Specified	Network Manager, Security
ground_base_url	String	256, TEXT	URL of the source/destination directory or file. This is the base URL. Absolute URL mentioned in UCF transferSettings.txt file should match with one of the ground_base_url list in HCF.	Field Not Specified	Data Transfer Function

Example:


```
{
 "whiteList": [
  {
   " description ": "LSAP Sync",
   " ground_base_url ": "https://aero.api-beta.honeywell.com"
  },
  {
   " description ": "Log Sync",
   " ground_base_url ": "https://aero.log-beta.honeywell.com"
  }
 ]
}
```


ID : SSS_4184

1AWObjectType : Description

Aircraft Wired Network Details (lans.txt) file defines the connection details for the wired Ethernet ports on the ADG-400. The first Ethernet port (ETH 1) is connected to the Avionics LAN on the aircraft. Only ports that are physically wired on the aircraft need to be defined. Unused ports should remain undefined.

Note that 172.20.201.xxx and 172.16.1.xxx are reserved by the WU and cannot be assigned to a wired Ethernet port.

An Isolation parameter is to specify whether that wired Ethernet port should be isolated from all other wired Ethernet ports and allowed access to the wireless interfaces. This parameter will help ensure that there is no access between the internet and the Avionics LAN. Only one port may have the Isolation parameter set to true.

ID : SSS_4185

1AWObjectType : Requirement

The lans.txt file **shall** be formatted in JSON format and contains an array of the following fields

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
eth	number	1, UNSIGNED, RANGE 1-5	number of the physical Ethernet port, 1 through 5	"1"	Network Manager
name	string	64, TEXT	name or description of the network that is connected to the Ethernet port	"Avionics LAN"	Network Manager
ip_address	string	64, IPADDRESS	IP Address of the Ethernet port	"192.168.200.6"	Network Manager
subnet	string	64, IPMASK	subnet mask	"255.255.0.0"	Network Manager
isolated	Boolean	N/A	flag to specify if the port is isolated from the other wired Ethernet ports and allowed access to wireless interface. Only one may be selected. True or false.	Field Not Specified	Network Manager, Security

Example of a lans.txt file:

```
{
"lans": [
        {
        "eth": 1,
        "name": "Avionics LAN"'
        "ip_address": "192.168.200.6"'
        "subnet": "255.255.0.0",
        "isolated": false
        }
        {
        "eth": 2,
        "name": "Crew LAN"'
        "ip_address": "172.20.200.6"'
        "subnet": "255.255.255.0",
        "isolated": true
  }
 ]
}
```

ID : SSS_4189

1AWObjectType : Requirement

The hardwaremode.txt file **shall** be formatted in JSON format and contains the following fields

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
ADG-400_hardware_pn	String	15, TEXT	ADG-400 hardware part number	"UNKNOWN"	Fault and Event Logging BITE

Example:

{

"ADG-400_hardware_pn ": "60006023-1000"

}

This parameter will control the hardware configurations available for various platforms based on the hardware part numbers. Refer SSS_645 for the various part numbers and their supported configuration.

ID : SSS_4190

1AWObjectType : Requirement

The softwareconfig.txt file **shall** be formatted in JSON format and contains an array of the following field.

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
adg_enabled_function	Array of string	SIZE 20, TEXT	This parameter is an array of strings which allows controlling various ADG-400 software functions. Each string represents one of the software function of the ADG-400.	[data_transfer_function, target_loading_epic, target_loading_a615a, target_loading_a615-3, a429_bus_recording, a429_bus_streaming, a717_bus_recording, ascb_bus_recording, ascb_bus_streaming, ascb_over_ethernet_recording, albf_function, connected_aircraft_service]	ADG Mode Controller
internet_time_url	string	256, TEXT	URL of the internet time server used by the ADG-400 Date/Time function when synching date/time with Internet Based Time server as preferred source.	time-a-g.nist.gov	Date/Time

ID : SSS_4191

1AWObjectType : Description

This file is a software function enabling text file (with filename softwareconfig.txt) defining the enable/disable requirements of the software function of the ADG-400. The ADG-400 system will have different virtual machines for keeping domain isolation for the enabled software functions. For example, one virtual machine for handling cabin traffic, other for handing file server and one for handling avionics domain. The ADG-400 system can shutdown these virtual machines if software function performed to be performed by virtual machines are disabled.

This adg_enabled_function parameter is an array of string. Each string is a ADG-400 software function name, which need to be enabled and supported by ADG-400 system.

Example:

{

"**adg_enabled_function**": ["data_transfer_function", "target_loading_epic", "target_loading_a615a", "target_loading_a615-3", "a429_bus_recording", "a429_bus_streaming", "a717_bus_recording", "ascb_bus_recording", "ascb_bus_streaming", "ascb_over_ethernet_recording", "albf_function", "connected_aircraft_service"],

"**intenet_teime_url**": "time-a-g.nist.gov"

}

ID : SSS_4193

1AWObjectType : Requirement

The nvmpartitionscale.txt file **shall** be formatted in JSON format and contains an array of JSON of the following fields

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
partition_name	String	PARTITION_NAME	NVM partition name	Field Not Specified	Software Modes of Operation
partition_size	number	5, UNSIGNED, RANGE 0-65535	Partition size	Field Not Specified	Software Modes of Operation
size_unit	String	ALLOWED_VALUES	Define the partition size unit. Allowed value is “GB”	Field Not Specified	Software Modes of Operation

Example:

```
{
"nvmpartitionscale": [
        {
        "partition_name": "Data Bus Recording",
        "partition_size": 300,
        "size_unit": “GB”,
        },
        {
        "partition_name": "DTF-A",
        "partition_size": 20,
        "size_unit": “GB”,
        },
        ]
}
```

nvmpartitionscale.txt file defines the partition size for all partition except primary operational image, secondary image and F&E.

ID : SSS_4682

1AWObjectType : Requirement

The aircraftWiring.txt file **shall** be formatted in JSON format and contains the following fields required for 615A self-loading and target loading.

TAG	Format	Description	Format	Character Maximum length, Type	Default	Functional Area
target_identifier	Array of JSON	"**target_hw_id**": Hardware ID. 15 characters maximum	String	15, TEXT	{ "aircraftwiring": { "target_identifier": { "target_hw_id": "Field Not Specified", "target_type_name": "Field Not Specified", "target_pos": "Field Not Specified", "literal_name": "Field Not Specified", "manufacture_code": "Field Not Specified", "targetIPAddress": "Field Not Specified", "a615_dlp_time_out": 13, "a615_dlp_retry_number": 1, "a615_tftp_time_out:": 2, "a615_tftp_retry_number": 1, "a615_wait": 50 "blockSizeOptionValue": 512, "portOptionValue": 0, "ephemeralPortA": 0, "ephemeralPortB": 0, "isDlAttachMsg": 0, "dataloadAttachedIP": 0, "dataloadAttachedPort": 0, "dlAttachedDisreteBit": 0 } } }	Self-loading Target Loading
		"**target_type_name**": Target Avionics name. 8 characters maximum	String	8, TEXT		
		"**target_pos**": Position of the Target (A or P domain).8 characters maximum	String	8, TEXT		
		"**literal_name**": Literal name of the target avionics. 20 characters maximum	String	20, TEXT		
		"**manufacture_code**": The Manufacturer's Code is an identification code assigned to each organization that develops aircraft software. 3 characters	String	3, TEXT		
		"**targetIPAddress**": the Target LRU IP Address	String	64, IPADDRESS		
		"**a615_dlp_time_out**": The DLP Time-Out is the time between the emission of the last packet of the initial TFTP (acknowledge) and the reception of the first packet of the secondary TFTP (write request)	number	13, UNSIGNED, RANGE 0-99		
		"**a615_dlp_retry_number**": DLP Retry Number is the number of times a TFTP file transfer is attempted again, after one initial failed TFTP file transfer.	number	1, UNSIGNED, RANGE 0-10		
		"**a615_tftp_time_out**": TFTP time-out is the time measured between the emission of one TFTP packet (whatever the type of TFTP packet) and the reception of the associated answer packet for the same TFTP exchange	number	2, UNSIGNED, RANGE 0-99		
		"**a615_tftp_retry_number**" The TFTP Retry Number is the number of times a TFTP exchange is attempted after one initial failed exchange	number	1, UNSIGNED, RANGE 0-10		
		"**a615_wait**": The ADG-400 (Data Loader) or the Target avionics can initiate a wait message. This message may be generated in response to any TFTP transfer request.	number	50, UNSIGNED, RANGE 0-99		
		"**blockSizeOptionValue**": The tftp block size supported for the LRU. Allowed values are 512, 1024, 1432, 2048, 4096, 8192	number	ALLOWED_VALUES		
		"**portOptionValue**": Port-option tftp server port value	number	5, UNSIGNED, RANGE 1-65535		
		"**isDlAttachMsg**": to indicate whether dl attach msg should be sent	Boolean	N/A		
		"**dataloadAttachedIP**": IP Address to send the dataload attached discrete	number	64, IPADDRESS		
		"**dataloadAttachedPort**": UDP port to send the dataload attached discrete	number	5, UNSIGNED, RANGE 1-65535		
		"**dlAttachedDisreteBit**": Discrete bit position in dataload attached msg	number	1, UNSIGNED, RANGE 0 to 255		

Example of aircraftwiring.txt:

{

"dataloading_info": {

"target_identifier": {

"target_hw_id": "Field Not Specified",

"target_type_name": "Field Not Specified",

```
                "target_pos": "Field Not Specified",
                "literal_name": "Field Not Specified",
                "manufacture_code": "Field Not Specified",
                "targetIPAddress": "Field Not Specified",
                "a615_dlp_time_out": 13,
                "a615_dlp_retry_number": 1,
                "a615_tftp_time_out:": 2,
                "a615_tftp_retry_number": 1,
                "a615_wait": 50
                "blockSizeOptionValue": 512,
                "portOptionValue": 0,
                "isDlAttachMsg": 0,
                "dataloadAttachedIP": 0,
                "dataloadAttachedPort": 0,
                "dlAttachedDisreteBit": 0
        }
}
}
```


Note: For self-loading, "target_hw_id" (HNIADG400), "a615_dlp_retry_number" and "a615_wait" parameters are used currently.

ID : SSS_5256

1AWObjectType : Requirement

The aircraftWiring.txt file **shall** be formatted in JSON format and contains the following fields required for A615-3 target loading.

TAG	Child Tag	Format	Description
wiringtype		String	Tag that identifies the specific wiring configuration.
LDRSystems		Array	Array of equipment which can be loaded and the Loader protocol to be used to load them
	target_type_name	String	System Name found in the Crate that identifies the system to be loaded. If the system name in the Crate does not match one of these records, the content of the Crate cannot be loaded. (e.g. FMS, MMR, DU). (in Crate.xml) Under the AssemblyItem->RelatedItem->there should be LDRSystem tag present.
	Target_pos	String	"L", "R", "C" designator. Used with the System name to report status on the Load operation. i.e. FMS-L, etc.
	LDRProtocol	Number	0 – A615A protocol 1 – A615-3 protocol 2 – EPIC protocol
	SAL	Number	Label number
	A429_channel_iface	String	A429 channel interface
	TxPort	String	Transmitter to map the A615-3 Tx data to for the Loader target. If missing, then transmitter is not supported
	RxPort	String	Receiver to map to the A615-3 Rx to for the loader target. If missing, then receiver is not supported
	Discrete	String	Discrete output to set for Loader Function Disc 1. Range is 1-5. If missing, then discrete out is not supported
	a429_to_etherenet	Boolean	use to enable UDP transfer of A429 bus data during A615-3 Loading, Default value is false.

Example:

"aircraftWiring":[

{

"wiringtype":"A300Type1",

"GPSRxPort":8,

"GPSRxHiSpd":true,

"WowDiscrPort":1,

"LDRSystems":[

{

"target_type_name":"FMS",

"target_pos":"L",

"LDRProtocol":1,

"SAL":300,

"A429_channel_iface":"core A429/A429 Module",

"Tx Port":"Tx1/Tx2",

"Rx Port":"Rx1/Rx2/Rx3/Rx4",

"Discrete":"Discrete Output1 through Discrete Output5",

"a429_to_etherenet":false

},

{

"target_type_name":"FMS",

"target_pos":"R",

"LDRProtocol":1,

"SAL":300,

"A429_channel_iface":"core A429/A429 Module",

"Tx Port":"Tx1/Tx2",

"Rx Port":"Rx1/Rx2/Rx3/Rx4",

"Discrete":"Discrete Output1 through Discrete Output5",

"a429_to_etherenet":false

}

ID : SSS_4222

1AWObjectType : Description

The ADG-400 HCF file contains the following certificates/keys.

Cert/Key name	Description	Default	Functional Area
config/HCFGroundEncryptKey.private	a private key (RSA 2048) provided by HON used to decrypt the data in the HCF.	Field Not Specified	Configuration Manager
config/HCFGroundEncryptKey.passphrase	a text file having the passphrase for the HCFGroundEncryptKey.private	Field Not Specified	Configuration Manager
config/HCFSigningCA.CER	CA (x509v3) provided by HON used to validate the Signing Public certificate which is received along with the signature file and data In the HCF	Field Not Specified	Configuration Manager
config/HCFSigningCA.CRL	CRL (x509v2) provided by HON for the certificate provided by HCFSigningCA.CER	Field Not Specified	Configuration Manager
radiusserver/radiusServerKey.p12	radiusServerKey.p12 is a public private key pair provided by HON PKI Provider for the WPA2/AES/Enterprise communication between mobile device and 1AW	Field Not Specified	Network Manager, Security
radiusserver/wifiClientCA.CER	wifiClientCA.CER is a CA(x509v3) provided by HON PKI Provider. Used to validate the mobile device client certificate	Field Not Specified	Network Manager, Security
radiusserver/wifiClientCA.CRL	wifiClientCA.CRL is a CRL (x509v2) provided by HON PKI Provider for the certificate provided by wifiClientCA.CER	Field Not Specified	Network Manager, Security
sslserver/standardServerSSLKey.p12	standardServerSSLKey.CER is a public private key pair provided by HON PKI Provider for the access point https communication protocol between Client application and ADG400.	Field Not Specified	Network Manager, Security
sslserver/standardClientSSLCA.CER	standardClientSSLCA.CER is a CA(x509v3) provided by HON PKI Provider. Used to validate the client certificate for https 2 way SSL handshake between Client application and ADG400.	Field Not Specified	Network Manager, Security
sslserver/standardClientSSLCA.CRL	standardClientSSLCA.CRL is a CRL (x509v2) provided by HON PKI Provider for the certificate provided by standardClientSSLCA.CER	Field Not Specified	Network Manager, Security

ID : SSS_132

1AWObjectType : Title

3.3.6.1.1.1.2 ACF Contents

ID : SSS_137

1AWObjectType : Description

There is one Aircraft Configuration File (ACF) per aircraft. The ACF contains information about the specific aircraft installation including a unique identifier which can be used to ensure that the ADG-400 is installed on the intended aircraft. The ACF file will be created by the customer using the Honeywell SIGNS server. Alternately, the engineering version of the ADG-400 Configuration Tool can be used to create an ACF (for development and integration purposes).

ID : SSS_138

1AWObjectType : Description

The ACF is signed only (not encrypted) and is signed using the HCF Signing Keys.

The ACF will be composed of a single archive file named sign_acf_N.zip, where “n” is a number that can be used for identification purposes (such as a software part number or version number).

ID : SSS_139

1AWObjectType : Description

Each text file in the ACF file is formatted in JSON format.

ID : SSS_4195

1AWObjectType : Requirement

The sign_acf_N.zip archive **shall** consist of the following files:

- ACFheader.txt
- aircraftSetting.txt

ID : SSS_4196

1AWObjectType : Requirement

The ACFheader.txt file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
part_number	String	15, TEXT	Part Number associated with the ACF container	Field Not Specified	Configuration Manager
version	String	6, UNSIGNED	Version number of the ACF format	"000000"	Configuration Manager
creation_date	String	19, TEXT	creation date and time in MM/DD/YYYY HH:MM:SS format	"01/01/2016 12:00:00"	Configuration Manager
digest	String	64, TEXT	SHA-256 digest of all of the files in the ACF excluding the header.txt in alphabetical order	""	Configuration Manager

Example:

{

"part_number": "60004873",

"version": "000005",

"creation_date": "01/01/2017 12:00:00",

"digest": "db8b794c7c214dbaea257f92a7218440aa61fe16c034df87d1cbb0d19dc0d"

}

ID : SSS_4197

1AWObjectType : Requirement

The ACF aircraftSetting.txt file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum length, Type	Description	Default	Functional Area
ac_tail_number	string	64, TEXT	tail number of the aircraft	"UNKNOWN"	BITE SSID Setup
ac_serial_number	string	64, TEXT	serial number of the aircraft	"UNKNOWN"	BITE
serial_number	string	15, TEXT	Serial number of the ADG-400	"UNKNOWN"	Fault and Event Logging
uuid	string	15, TEXT	Universally Unique Identifier of the ADG-400. The UUID will be generated by SIGNS or by the ADG-400 Configuration Tool using the ADG-400's hardware version and serial number. For the ADG-400, the UUID will be "G7222-{dash number}-{serial number}", where {dash number} is currently "001" to reflect the ADG-400 and WU tray hardware and {serial number} is the ADG-400 serial number	NULL (Empty Field)	Fault and Event Logging

Example of an aircraftSetting.txt file:

```
---
  {
        "ac_tail_number": "N1235",
        "ac_serial_number": "EMB1234",
        "serial_number": "38",
        "uuid": "G7222-001-38"
  }
```

ID : SSS_133

1AWObjectType : Title

3.3.6.1.1.1.3 UCF Contents

ID : SSS_140

1AWObjectType : Description

The UCF contains features for customization of the ADG-400 operation for an entire fleet of aircraft. Thus, the same UCF may be used for an entire aircraft fleet. The UCF file is created by aircraft manufactures and airlines via the Honeywell SIGNS Web portal itself or by downloading the Honeywell provided desktop tool (WCT) from the SIGNS Web portal to a Windows PC. The UCF is encrypted and digitally signed for security purposes when received by aircraft manufactures and airlines from the Honeywell.

ID : SSS_141

1AWObjectType : Description

The UCF can also be secured under the responsibility of the aircraft manufactures and airlines. In this case, UCFs will be encrypted and digitally signed for security purposes by using an aircraft manufactures

and airlines Certificate Authority (CA) rather than by the Honeywell CA. In this case the tools to generate the UCF will be provided by Honeywell but the actual security certificates will be provided into the tool by the aircraft manufacturers or airlines Certificate Authority (CA) chain of trust.

ID : SSS_142

1AWObjectType : Description

The UCF will be composed of a single archive file named:

- key_enc_sign_ucf_N.zip if current certificates and keys are used, where “n” is a number that can be used for identification purposes (such as a software part number or version number).

- key_enc_sign_ucf_000000_factory.zip if factory default certificates are used.

In addition, both key_enc_sign_ucf_N.zip and key_enc_sign_ucf_000000_factory.zip files may be placed in a single zip file named gs_key_enc_sign_ucf_60004873_N.zip and uploaded to the ADG-400.

ID : SSS_143

1AWObjectType : Description

Each text file in the UCF file is formatted in JSON format.

ID : SSS_4198

1AWObjectType : Requirement

The enc_sign_ucf_N.zip archive file in the ADG-400 **shall** contain the following files:

- UCFheader.txt
- cellularSetting.txt
- countrySetting.txt
- wlanSetting.txt
- transferSetting.txt
- apModeSetting.txt
- airportSetting.txt
- airportMaster.txt
- aircraftSetting.txt

- fleetSetting.txt
- lanTestSetting.txt
- debug.txt
- httpsServerSetting.txt
- config/UCFGroundEncryptKey.private
- config/UCFGroundEncryptKey.passphrase
- config/UCFSigningCA.CER
- config/UCFSigningCA.CRL
- sslserver/standardServerSSLKey.p12
- sslserver/standardClientSSLCA.CER
- sslserver/standardClientSSLCA.CRL
- sslclient/standardClientSSLKey.P12
- sslclient/standardServerSSLCA.CER
- radiusserver/radiusServerKey.p12
- radiusserver/wifiClientCA.CER
- radiusserver/wifiClientCA.CRL
- wificlient/wifiHotSpotCA.CER
- wificlient/wifiHotSportKey.p12

ID : SSS_4199

1AWObjectType : Requirement

The UCFheader.txt file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum Length, Type	Description	Default	Functional Area
part_number	String	15, TEXT	Part Number associated with the UCF container	Field Not Specified	Configuration Manager
version	String	6, UNSIGNED	Version number of the UCF format	"000000"	Configuration Manager
creation_date	String	19, TEXT	creation date and time in MM/DD/YYYY HH:MM:SS format	“01/01/2016 12:00:00”	Configuration Manager
digest	String	64, TEXT	SHA-256 digest of all of the files in the UCF excluding the header.txt in alphabetical order	“”	Configuration Manager

Example UCFheader.txt:


```
{
    "part_number": "60004872",
    "version": "000001",
    "creation_date": "01/15/2017 09:15:01",
    "digest": "5e7fb324d9c54ee0f61a3123648e4ff9724d452bba89edde884bb67a15e65399"
    }
```


ID : SSS_4208

1AWObjectType : Requirement

The access point configuration apModeSetting.txt file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum Length, Type, Range	Description	Default	Functional Area
ssid	string	64, TEXT	Defines the SSID format of the Access Point. The Tail Number is added to this string to form the actual SSID so that each aircraft has a unique SSID. The characters “<ac_tail_num>” in the string are replaced with the aircraft tail number.	No Factory Default UCF	SSID Setup
broadcast_ssid	Boolean	N/A	Flag specifying whether to broadcast the SSID of the Access Point. Allowed values are "true" or "false"	No Factory Default UCF	SSID Setup
wifiClient_ca_cert_filename	string	64, TEXT	Path of the certificate which is Used to validate the mobile device client certificate for WPA2/AES/Enterprise communication between mobile device and OBS.	No Factory Default UCF	Network Manager
wifiClient_crl_filename	string	64, TEXT	Path of the CRL for the wifi client ca certificate	No Factory Default UCF	Network Manager
radiusServer_p12_filename	String	64, TEXT	Path of the raidus server p12 file which is used by raidus server for WAP.	No Factory Default UCF	Network Manager
radiusServer_p12_passphrase	String	64, TEXT	Passphrase for radius server p12 file.	No Factory Default UCF	Network Manager

Example of an apModeSetting.txt file:

{

"ssid": "<ac_tail_num>_ADG400_AP",

"broadcast_ssid": false,

"wifiClient_ca_cert_filename": "radiusserver/wifiClientCA.cer",

"wifiClient_crl_filename": "radiusserver/wifiClientCA.crl",

"radiusServer_p12_filename": "radiusserver/ radiusServerKey.p12",

"radiusServer_p12_passphrase": "qwetwiuet219898kjljklkjl"

}

ID : SSS_4209

1AWObjectType : Description

The apModeSetting.txt file will contain the file name of the server-side certificate to be used in creating of the pair-wise certificates between remote terminal devices and the ADG-400. The remote terminals will request network access using IEEE802.1x/EAP-TLS.

If the ADG-400 will act as an access point, the Access Point Device Certificates and Private Key must be included in the ucf_N.zip file along with the configuration files.

ID : SSS_4200

1AWObjectType : Requirement

The cellularSetting.txt file **shall** be formatted in JSON format and contains the following fields for each SIM Card installed:

TAG	Format	Character Maximum Length, Type or Allowed Values	Description	Default	Functional Area
sim	number	ALLOWED_VALUES	Number of the SIM Card slot that contains a SIM Card. Allowed values are "1", "2".	No Factory Default UCF	Network Manager
apn	string	256, TEXT	access point name (APN) of the cellular provider	No Factory Default UCF	Network Manager
username	string	256, TEXT	Username required for connecting to the APN.	No Factory Default UCF	Network Manager
password	string	256, TEXT	Password required for connecting to the APN.	No Factory Default UCF	Network Manager
default	Boolean	N/A	This flag specifies whether or not the SIM Card will be used for the initial cellular connection attempt (for gathering time and country information). Only one SIM Card will be true. Allowed values are "true" or "false".	No Factory Default UCF	Network Manager

Example of a cellularSetting.txt file:

```
{
        "cellularSetting": [{
                        "sim": 1,
                        "apn": "broadband",
                        "username": "",
                        "password": "",
                        "default": true
                },
                {
```

```
                "sim": 2,
                "apn": "m2m.com.attz",
                "username": "",
                "password": "",
                "default": false
            }
        ]
}
```

ID : SSS_4201

1AWObjectType : Requirement

The countrySetting.txt **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Array Size, Allowed Value or Type	Description	Default	Functional Area
country_code	String	COUNTRY CODE	ISO 3166-1 alpha-3 Country Code	No Factory Default UCF	Network Manager
wifi_channel	number	WIFICHANNEL	Identifies which Channel to use for the Hot Spot when at this location. If value is blank or field is not included, then the fleetsettings.txt channel is used.	No Factory Default UCF	Network Manager
cellular_mode_enable	boolean	N/A	toggle cellular on and off	No Factory Default UCF	Network Manager
access_mode_enable	boolean	N/A	toggle access point on and off	No Factory Default UCF	Network Manager
wifi_mode_enable	boolean	N/A	Toggle client Wi-Fi on and off	No Factory Default UCF	Network Manager
network_priority	Array of String (optional)	SIZE 3, ALLOWED_VALUES	Ordered list of networks to use to connect to ground station when at this location. Allowed values are “Wi-Fi”, “Cellular1”, “Cellular2” or “”. If a network is not listed, it is not used at this location. If value is blank or field is not included, then the fleetsettings.txt priority is used	No Factory Default UCF	Network Manager

Example

```
{
    "countrySetting": [{
```

```
                "country_code": "USA",
                "wifi_channels": 6,
                "cellular_mode_enable": true,
                "access_mode_enable": true,
                "wifi_mode_enable": true,
                "network_priority": [
                        "cellular1",
                        "wifi", "Cellular2"
                ]
        }]
}
```


ID : SSS_4202

1AWObjectType : Commentary

The countrySetting.txt file defines the country preferences on cellular1, WiFi and cellular2 connections. When the ADG-400 starts up, a cellular radio (with a SIM card installed) will connect to a base station and obtain the Mobile Country Code (MCC). The corresponding ISO country code will be identified from countryMapping.txt. The ISO country code can then be used to determine which SIM card to be used for the detected location.

If a country is located in this file, then the access point channel and network priority are used as default values. If no country code record is found for the current location, then the client WiFi and Cellular radios are disabled and the access point WiFi is set to the lowest power setting.

The countrySetting.txt file is formatted in JSON format and contains an array of the following fields (one record per country_code). If a country is located in this file, then the access point channel and network priority are used as default values. If no country code record is found for the current location, then the client Wifi and Cellular radios are disabled and the access point Wifi is set to the lowest power setting.

ID : SSS_4203

1AWObjectType : Requirement

The wlanSetting.txt file **shall** be formatted in JSON format and contains the following fields

TAG	Format	Character Maximum Length, Type	Description	Default	Functional Area
Description	String	256, TEXT	Short textual description of the wireless network for readability purposes. This parameter is optional.	No Factory Default UCF	N/A
ssid	String	64, TEXT	SSID of the Wi-Fi network. Note that the first group will be considered the highest priority. The last group will be considered the lowest priority.	No Factory Default UCF	Network Manager
Passphrase	String	64, TEXT	Pre-shared key passphrase for the network (if PSK is used)	No Factory Default UCF	Network Manager
airport_code	String (optional)	AIRPORTCODE	ICAO	No Factory Default UCF	Network Manager
country_code	String (Optional)	COUNTRYCODE	ISO 3166-1 alpha-3 Country Code	No Factory Default UCF	Network Manager
ca_cert_filename	String	64, TEXT	Filename of the wi-fi network CA certificate. This file will be included along with the text files as part of the UCF.	No Factory Default UCF	Network Manager
p12_filename	String	64, TEXT	Filename of the Certificate/Key pair for the network (if TLS/EAP is used)	No Factory Default UCF	Network Manager
p12_passphrase	String	64, TEXT	Passphrase for the P12 file (if TLS/EAP is used)	No Factory Default UCF	Network Manager

Example:


```
{
"wlanSetting": [
{
    "description": "UPS hanger",
    "ssid": "ac_hanger",
    "passphrase": "baadbeef",
    "airport_code": "KPHX",
    "country_code": "USA",
    "ca_cert_filename": "",
    "p12_filename": "",
```

```
    "p12_passphrase": ""
},
{
    "description": "UPS hanger",
    "ssid": "ac_hanger",
    "passphrase": "baadbeef",
    "airport_code": "KPHX",
    "country_code":"",
    "ca_cert_filename": "",
    "p12_filename": "",
    "p12_passphrase": ""
},
{
    "description": "JFK Gate A1",
    "ssid": "gatelink",
    "passphrase": "",
    "airport_code": "",
    "country_code": "",
    "ca_cert_filename": "gatelink.ca",
    "p12_filename": "gatelink_client.p12",
    "p12_passphrase": "p@ssw0rd"
 }
]
}
```


ID : SSS_4204

1AWObjectType : Description

The wlanSetting.txt file defines the wireless networks that the ADG-400 will attempt to connect to when the Aircraft is on-ground. It is assumed that the wireless network will use either WPA-2 PSK or WPA-2 EAP/TLS authentication. If a Client Certificate and Private Key is required for connection to a wireless

network (as with EAP/TLS authentication), the P12 file containing the Client Certificate and Private Key must be included in the ucf_N.zip file along with the configuration files.

Note:

1. if “passphrase” fiield in wlans.txt is given then ‘WPA2 Personal’ protocol is used for Wi-Fi connection with given “passphrase”.

2. If “passphrase” field in wlans.txt is not given then ‘WPA2 Enterprise’ protocol is used for Wi-Fi connection with given "p12_filename" and "p12_passphrase".

3. airport_code and country_code are optional parameters. If values are set for these optional parameters then SSID is specific to particular airport/country.

ID : SSS_4205

1AWObjectType : Requirement

The transferSetting.txt file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum Length, Type, Range, Allowed Value	Description	Default	Functional Area
Description	string	256, TEXT	Short textual description of the transfer rule for readability purposes. This parameter is optional.	No Factory Default UCF	Data Transfer Function
ground_auth_url	string	256, TEXT	URL for authorization server which would provide access token for resource server		
ground_resource_url	string	256, TEXT	URL of the source directory or file. This is the primary location.		
ground_client_id	string	64, TEXT	client id required to access the source server when security type is oauth2.0		
ground_client_secret	String	256, TEXT	client secret required to access the source server when security type is oauth2.0		
device_resource_url	string	256, TEXT	URL of directory or file on device. This is the primary location.		
SSLclient_ca_cert_filename	string	64, TEXT	Used to validate the server certificate for https 2 way SSL handshake between ground server and ADG-400.		
SSLclient_p12_filename	string	64, TEXT	Public private key pair for the access point https communication protocol between ground server and ADG-400.		
SSLclient_p12_passphrase	string	64, TEXT	Passphrase for the SSL client P12 file.		

ID : SSS_4207

1AWObjectType : Description

The transferSetting.txt file defines outbound/inbound transfers such as databases from a ground server to the aircraft electronics, or maintenance information from aircraft electronics to ground servers. Transfer of ADG-400 log files to a ground server may also be defined in transfers.txt. The ADG-400 Configuration Tool will automatically set the source information for transfer of ADG-400 log files.

If a Client Certificate and Private Key is required for authentication with a server, the certificate file and key file must be included in the ucf_N.zip file along with the configuration files.

Example:


```
{
  "transferSetting": [
    {
      "description": "Database inbound job",
      "ground_auth_url": " https://aero-signs.api-beta.honeywell.com/v2/oauth2/client_credential/accesstoken?grant_type=client_credentials",
      "ground_resource_url": " https://aero-signs.api-beta.honeywell.com/v2/signs/1p1/device/1aw/{ac_tail_num} ",
      "ground_client_id": "xyz123",
      "ground_client_secret": "12aqwq3qr32",
      "device_resource_url": WDLS,
      “SSLclient_ca_cert_filename”: “standardServerSSLCA.CER”,
      “SSlclient_p12_filename”: “standardClientSSLKey.P12”,
      “SSLclient_p12_passphrase”:” qwetwiuet219898kjljklkjl”
    }
  ],
"transferSetting": [
    {
      "description": "Database inbound job",
      "ground_auth_url": " https://aero-signs.api-beta.honeywell.com/v2/oauth2/client_credential/accesstoken?grant_type=client_credentials",
      "ground_resource_url": " https://aero-signs.api-beta.honeywell.com/v2/signs/1p1/device/1aw/{ac_tail_num} ",
      "ground_client_id": "xyz123",
      "ground_client_secret": "12aqwq3qr32",
      "device_resource_url": CRL,
      “SSLclient_ca_cert_filename”: “standardServerSSLCA.CER”,
      “SSlclient_p12_filename”: “standardClientSSLKey.P12”,
```

“SSLclient_p12_passphrase”:” qwetwiuet219898kjljklkjl”

}

]

}

ID : SSS_4206

1AWObjectType : Requirement

The httpsServerSetting.txt file **shall** be formatted in JSON format and contains the following fields

TAG	Format	Character Maximum Length, Type	Description	Default	Functional Area
SSLserver_ca_cert_filename	String	64, TEXT	Used to validate the client certificate for https 2 way SSL handshake between mobile device and OBS.	No Factory Default	Data Transfer Function
SSLserver_crl_filename	String	64, TEXT	CRL for the SSL ca certificate	No Factory Default	Data Transfer Function
SSLserver_p12_filename	String	64, TEXT	Public private key pair for the https communication protocol between mobile device and OBS.	No Factory Default	Data Transfer Function
SSLserver_p12_passphrase	String	64, TEXT	Passphrase for the SSL server P12 file.	No Factory Default	Data Transfer Function

ID : SSS_4194

1AWObjectType : Description

Example of httpsServerSetting.txt file

{

"httpsServerSetting": [

{

"SSLserver_ca_cert_filename": "sslserver/standardClientSSLCA.CER",

"SSLserver_crl_filename": "sslserver/standardClientSSLCA.CRL",

"SSlserver_p12_filename": "sslserver/standardServerSSLKey.P12",

"SSLserver_p12_passphrase": "qwetwiuet219898kjljklkjl"

}

]

}

ID : SSS_4217

1AWObjectType : Requirement

The Wireless Transfer Queue configuration queue.txt file **shall** be formatted in JSON format and contains the array of following fields:

TAG	Format	Character Maximum Length, Type, Range or Allowed Valuess	Description	Default	Functional Area
priority_queue	string	ALLOWED_VALUES	Gives all bandwidth to a single queue for a specified period of time. Allowed values are "inbound", "outbound", or "both".	No Factory Default UCF	Data Transfer
priority_period	number	3, UNSIGNED, RANGE 0-600	Maximum time that the priority queue will have all bandwidth. [0:600] seconds	No Factory Default UCF	Data Transfer
outbound_repeat_delay	number	5, UNSIGNED, RANGE 0-65535	Period of time (in seconds) that the ADG-400 will delay before scanning aircraft server folders again. [0:65535] seconds	No Factory Default UCF	Data Transfer

Example of an queue.txt file:

{

" queue.txt": [

{

“priority_queue”: "outbound",

"priority_period”: 300,

"outbound_repeat_delay": 1

}

]

}

ID : SSS_4218

1AWObjectType : Description

The queue.txt file defines the Wireless Transfer Queue details. By default, the ADG-400 will process the inbound and outbound transfer queues at the same time. If required, these details will allow a user to assign all of the available wireless bandwidth to a single transfer queue (inbound or outbound) for a specified amount of time. When priority is defined for a particular queue this will not limit the wired LAN side transfers sub-jobs. For example if the wireless interface priority is set to the inbound queue for 4 minutes the outbound queue will still be collecting the source files from the avionics servers but will not be permitted to start the second half of the transfer over the wireless interface until the 4 minutes has expired or the inbound queue has no additional files to transfer. In the counter example if the wireless interface priority is set to outbound the inbound queue would only be able to finish up any previously unfinished sub-jobs from the ADG-400 to the avionics file servers until the following conditions occur: 4 minute timer expires, there is no additional files to be transferred per the outbound rules, or the outbound tasks have all completed and the outbound queue is in its outbound_repeat_delay period.

The queue.txt file will also define the outbound repeat delay. The outbound queue will continue to look for new data from all of the avionics file servers while the aircraft is on ground and powered up. This parameter defines the amount of time to wait before re-processing the outbound queue.

The priority_queue parameter is used to specify which queue has the first priority. If this parameter is “both” then the inbound and outbound queues will be processed at the same time. If only cellular connection is available, it may be necessary to give the outbound queue (from ADG-400 to Ground) priority to ensure that large maintenance files can be transferred to the Ground server after each flight leg.

This outbound_repeat_delay parameter will specify the amount of time (in seconds) that the ADG-400 will delay before repeating a scan of the aircraft’s server folders and transferring any new files or data in the outbound queue.

ID : SSS_4219

1AWObjectType : Requirement

The aircraftSetting.txt file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum Length, Type, Range	Description	Default	Functional Area
ac_tail_number	String	16, TEXT	A/C Tail number		
ac_serial_number	String	16, TEXT	A/C Serial number		
serial_number	String	15, TEXT	Serial number of ADG400		
uuid	String	15, TEXT	Universally Unique Identifier of the ADG-400. For the ADG-400, the UUID will be "G7222-{dash number}-{serial number}", where {dash number} is currently "001" to reflect the ADG-400 and WU tray hardware and {serial number} is the ADG-400 serial number		Faults & Event Logging
rt_wifi_channel	Number	ALLOWED_VALUES	Wi-Fi channel to be used when the ADG-400 is in AP Mode ("1", "6" or "11")		Network Manager
rt_retry_attempts	Number	3, UNSIGNED, RANGE 1-100	Number of times to retry a connection to a remote terminal using the one-time password. Max number of connection attempts ("1" to "100")		Network Manager
rt_backoff_period	Number	120, UNSIGNED, RANGE 1-600	Time in seconds to wait before trying a new remote terminal connection if the previous N attempts failed. The parameter defines the amount of time that a specific IP Address must wait before being allowed to connect once the maximum number of retry attempts has failed.		Network Manager
rt_max_clients	Number	3, UNSIGNED, RANGE 1-100	Maximum number of 802.11 clients allowed.		Network Manager
rt_connect_fault_limit	Number	3, UNSIGNED, RANGE 1-100	limit the number of erroneous connections to a connect fault value specified		Network Manager
rt_auth_fault_timer_limit	Number	3, UNSIGNED, RANGE 1-100	Timer Limit in seconds within which the authentication of remote client for the requesting CA service is expected to be completed for which the challenge is provided.		Network Manager
rt_blacklist_device_limit	Number	3, UNSIGNED, RANGE 1-100	Disable DHCP services if the number of black listed devices exceeds the threshold.		Network Manager
ext_app_timeout_limit	Number	3, UNSIGNED, RANGE 60-600	The time interval between two heartbeat messages from the External Apps to ADG-400 should not exceed the external App timeout.		Network Manager
ap_mode_enable	Boolean	N/A	access point enable flag ("true" or "false")		Network Manager
wifi_mode_enable	Boolean	N/A	wi-fi enable flag ("true" or "false")		Network Manager
cell_mode_enable	Boolean	N/A	cellular enable flag ("true" or "false")		Network Manager
A429_channel_rec_enable	Boolean	N/A	A429 channel recording enable flag ("true" or "false")		Data Bus Recording
Passphrase	String		Access Point passphrase key		Network Manager
daisy_chain_enabled	Boolean	N/A	Whether daisy chain is enabled or not for this aircraft.		AVIoP
satcom_type	String	ALLOWED_VALUES	Whether satcom is enabled or not, if enabled which satcom type it is mentioned here like SDR, AOC, SKYWAN (or) None		AVIoP
vpn_enabled	Boolean	N/A	Whether VPN is enabled or not for this aircraft		AVIoP
wifi_in_air	Booelan	N/A	By default, wifi_in_air will be enabled(true).		Network Manager
heartbeat_timeout	Number	3, UNSIGNED, RANGE 0 - 999	The amount of time (in seconds) that the ADG-400 should wait before terminating a connection with a mobile device if no heartbeat is detected. Note that when a mobile device is connected to the ADG-400 a heartbeat signal will be maintained between the ADG-400 and mobile device so that the ADG-400 can detect if the device is out of range, powered off, etc.		Network Manager
Platform_type	String	64, TEXT	Which platform this ADG is interacting like epic, legacy (or) none.		
A429_Channel_GPS	String	ALLOWED_VALUES	Determine on which A429 channel GPS data is available for location, Date and Time. Allowed values are "Disable", "Channel 1", "Channel 2", "Channel 3", "Channel 4"		Network Manager,DateTime
auto_network_link	Boolean	N/A	Automatic network selection for automatic link negotiation based on network priority. It will be useful when RADIO SELECT discrete is not wired		Network Manager
wap_anntenna_config	String	ALLOWED_VALUES	To select antenna for WAP operation. Allowed values are "front", "rear" (or) "discrete"		Network Manager
wap_cable_attenuation	Number	2, UNSIGNED, RANGE 1-10	This value is for each aircraft module, it will be 0 – 10 & it is pre-populated based on aircraft type.		Network Manager
EnhanceMode	Boolean	N/A	Whether to enable enhance mode for display of directory listing.		FTP Server
adg_gui_username	String	16,TEXT	This value is used to access adg gui when the user is connected through wired connection.		GUI Webserver
adg_gui_password	String	16,TEXT	This value is used to access adg gui when the user is connected through wired connection.		GUI Webserver
chartEnable	Boolean	N/A	Whether to enable the charts feature for ADG-400		FTP Sever

ID : SSS_4220

1AWObjectType : Guidance

Example:

```
{
"aircraftSetting": [
    {
    "uuid": "G7222-001-38",
    "serial_number": "38",
    "ac_tail_number": "N1235",
    "ac_serial_number": "EMB1234",
    "ap_mode_enable": false,
    "rt_wifi_channel": 6,
    "rt_retry_attempts": 3,
    "rt_backoff_period": 120,
    "rt_max_clients": 8,
    "wifi_mode_enable": true,
    "cell_mode_enable": true,
    "QAR_framerate": 32,
    "QARbiphaseformat": true,
    "Passphrase": "asdfghjklzxcvbnmqwertyuiop",
    "rt_connect_fault_limit": 3,
    "rt_auth_fault_timer_limit": 3,
    "rt_blacklist_device_limit": 3,
    "ext_app_timeout_limit": 2,
    "A429_channel_rec_enable": true,
    “daisy_chain_enabled”:true,
    “satcomType”: “SDR”,
    “vpn_enabled”:true,
    “wifi_in_air”:true,
    “heartbeat_timeout”:10,
    “platform_type”:”epic”,
```

“A429_Channel_GPS”:” disable”,

“auto_network_link:true,

“wap_antenna_config”:”rear”

“wap_cable_attenuation”:7,

"EnhanceMode":"true",

"adg_gui_username":"superuser",

"adg_gui_password":"password",

"ChartEnable":"false"

}

]

}

ID : SSS_4212

1AWObjectType : Requirement

The fleetsetting.txt file **shall** be formatted in JSON format and contains the following fields

TAG	Format	Character Maximum Length, Type, Range, Allowed Values	Description		Functional Area
allow_file_ext	Array of Strings	256, TEXT	List of files that are allowed to be loaded on to ADG-400. This list apply to any application which downloads new files to the ADG-400.	NoFactoryDefault	Data Transfer
wifi_mode_enable	Boolean	N/A	Wi-Fi enable flag ("true" or "false") If false, then the Client Wi-Fi Radio is always disabled and no attempts to sync via Client Wi-Fi are made.		Network Manager
cell_mode_enable	Boolean	N/A	Cellular enable flag ("true" or "false"). If false, then the Cellular Radio is always disabled and no attempts to sync via Cellular are made.		Network Manager
airport_radius	number	3,UNSIGNED, RANGE 1-999	Radius (miles) from airport lat/long. If current GPS Position is within this distance of an airport location, then that airport is the current location of the ADG-400.		Network Manager Location
recordings_retention_period	number	3, UNSIGNED, RANGE 1-365	Number of days to retain the QAR log files, ASCB, Ethernet and A429 channel recordings after creation if not successfully uploaded to the ground server		Data Bus Recordings
log_file_retention_period	number	3, UNSIGNED, RANGE 1-365	Number of days to retain the Activity/Security log files after creation if not successfully uploaded to the ground server		Data Transfer
ap_mode_enable	Boolean	N/A	access point enable flag ("true" or "false")		Network Manager
wifi_channel	number	WIFICHANNEL	Wi-Fi channel to use for the Hot Spot.		Network Manager
queue_freq	Number	5, UNSIGNED, RANGE 1 - 65536	How often to "phone home" in seconds. This is a file check frequency for inbound/outbound between Ground to ADG-400 (1-65536 secs). It will be repeated after complete queue operation is complete		Data Transfer
status_queue_freq	number	5, UNSIGNED, RANGE 1 - 65536	This is the frequency to communicate the inbound and the outbound activity statuses from ADG-400 to Ground (1-65536)		Data Transfer
file_check_retry_freq	number	2, UNSIGNED, RANGE 1 - 59	How soon to retry if file check is unsuccessful in seconds		Data Transfer
file_check_retry_attempts	number	2, UNSIGNED, RANGE 1 - 99	How many times to retry file check		Data Transfer
area_network_priority	JSON	SIZE 6, ALLOWED_VALUES	Which network should be taken as priority based on the area. If the network type is not defined for area, then it should not be used for that area.		Data Transfer
network_priority	Array of String	64, TEXT	Ordered list of network to use to connect to ground station when at this location. Allowed Values are Wifi, Cellular and Satcom. If a network is not listed, it is not used.		Network Manager
area_priority	Array of String	SIZE 6, ALLOWED_VALUES	Order in which various files categories are Sync'ed to the Ground Server. Allowed values are "LOGS", "WDLS", "QAR", "CONFIG", "FMAN", "MMAN". Should be an array of 6 values. If a category is missing, it is not synced to the ground.		Data Transfer
sslClientValParameters	string	64, TEXT	SSL client parameters needs to be validated are captured here. Note: Web Server should use these parameters (regex search) to accept/reject Client application 2 WAY SSL handshake after successful trust anchor. Example regex: ^.*O=(Honeywell International Inc|UPS)\/OU=62202\/OU=PROD\/CN=62202_.*\/.*$		Data Transfer, Security
Delete	Boolean	N/A	It will set as true by default and it is applicable only for outbound operations.		Data Transfer, Security
FIFO	Boolean	N/A	It will be set to true by default.		Data Transfer, Security
wifiClientValParameters	string	64, TEXT	wifi client parameters needs to be validated are captured here. Note: RADIUS Server should use these parameters (regex search) to accept/reject Wi-Fi client after successful trust anchor.		Data Transfer, Security
auto_getlog_enable	Boolean	N/A	Copy the activity log to SD card slot1/status/actStatus.log when the operational mode transitions from air to ground, when enabled. ("true" or "false")		Data Transfer
wifi_in_air	Booelan	N/A	By default, wifi_in_air will be enabled(true)		Network Manager
heartbeat_timeout	Number		ADG400 heart beat timeout.		Network Manager

ID : SSS_4214

1AWObjectType : Description

Example of the fleetsetting.txt file:


```
{
  "allow_file_ext": ["zip", "cer", "crl", "000", "BIN" ],
  "wifi_channel": 6,
  "ap_mode_enable": true,
  "wifi_mode_enable": false,
  "cell_mode_enable": false,
  "queue_freq": 3600,
  "status_queue_freq": 60,
  "file_check_retry_freq": 300,
  "file_check_retry_attempts": 3,
  "airport_radius": 5,
  "log_file_retention_Period": 60,
  "area_network_priority": {"wdls": ["cellular1","wifi","satcom"], "logs": ["wifi", "cellular"]},
"network_priority": [ "cellular1", "wifi", "satcom"],
  "area_priority": ["WDLS", "MMAN", "FMAN", "LOGS", "DATA", "CONFIG" ],
"sslClientValParameters": "^.*O=(Honeywell International
Inc)/OU=62202/OU=PROD/CN=62202_.*/.*$",
"wifiClientValParameters": "^.*O=(Honeywell International
Inc)/OU=62202/OU=PROD/CN=62202_.*/.*$",
"delete":true,
"FIFO":true,
"auto_getlog_enable": true,
"wifi_in_air":true,
"heartbeat_timeout":10
}
```


ID : SSS_4192

1AWObjectType : Requirement

The aircraftSetting.txt file and fleetsetting.txt file **shall** contain the following additional fields for Data Bus Recording:

ascb_data_bus_recording	Boolean	ascb bus recording enable flag. Allowed values are ("true" or "false").	N/A	No Factory Default UCF	Data Bus Recording
ascb_over_eth_recording	Boolean	ascb over the ethernet bus recording enable flag. Allowed values are ("true" or "false").	N/A	No Factory Default UCF	Data Bus Recording
ttgbe_over_eth_recording	Boolean	ttgbe bus recording enable flag. Allowed values are ("true" or "false").	N/A	No Factory Default UCF	Data Bus Recording
cmc_data_recording	Boolean	cmc bus recording enable flag. Allowed values are ("true" or "false").	N/A	No Factory Default UCF	Data Bus Recording
A429_channel_Info	Array of JSON	"A429_channel_rec_enable": recording enable flag. Allowed values are ("true" or "false").	Boolean	No Factory Default UCF	Data Bus Recording
		"A429_channel": A429 channel number. Valid values are (1, 2, 3, and 4).	1, UNSIGNED, RANGE 1 – 4		
		"A429_channel_name": Name of the A429 channel.	16, TEXT		
		"A429_channel_iface": Indicate whether A429 channel interface with Core A429 module or A429 RMI module. Allowed values are ("Core A429" and "A429 Module")	ALLOWED_VALUES		
		"A429_hi_rate": Bus rate flag. Allowed values are ("true" or "false"), Default is true (High speed)	Boolean		
		"A429_channel_tx": Transmit/Receive flag. Allowed values are ("true" or "false")			
A429_record_filter	JSON	"**record_filter**" — "A429_channel": A429 channel to be recorded, valid values are (1, 2, 3, 4).	1, UNSIGNED, RANGE 1 – 4	No Factory Default UCF	Data Bus Recording
		"A429_channel_iface": Indicate whether A429 channel interface with Core A429 module or A429 RMI module. Allowed values are ("Core A429" and "A429 Module")	ALLOWED_VALUES		
		"A429_label": Array of A429 labels to be recorded. Record all labels if field is empty or not defined. "	SIZE 377, 3,OCTAL, RANGE 0-377		
		A429_parity": true/false,	Boolean		
		"A429_ssm": SSM is the sign/status, valid values are (0, 1, 2, and 3).	1, UNSIGNED, RANGE 1 – 4		
		"A429_record_rate: recording rate in ms. Allowed values are {100, 250, 500, 1 sec, 2 secs, 6 sec, 15 sec, 30 sec, 1 min and on change}. Default 1 sec.	ALLOWED_VALUES		
		"**A429_passcode**": When A429 recordings are compressed into ZIP format, if the A429_passcode is supplied, it will be encrypted using this passcode.	64, TEXT		
		"**A429_file_name**": Naming convention to be followed for A429 record files. Starts with A429 and ends with ADG. Variable fields are: <Tail>, filled in with the Tail of the aircraft, <FlightLeg>, filled in with flight leg, <DateTime>, filled in with the date/time the flight ended in the format YYYYMMDD_HHMMSS, data type, the Airport ICAO code where the flight ended and the <Gateway>. Example: "A429_<Tail>_<FlightLeg>_<DateTime>_GAMA_<ICAO>_ADG.zip"	64, TEXTs		

ID : SSS_4667

1AWObjectType : Requirement

The aircraftSetting.txt file and fleetsetting.txt file **shall** contain the following additional fields for Data Bus Recording:

A717_QAR_rec	JSON	**“A717_QAR_rec**”: A717 QAR channel recording enable flag. Allowed values are ("true" or "false").		Boolean	No Factory Default UCF	Data Bus Recordin g
		“A717 _chann el_info ”	**“A717_framerate”:** Defines QAR bus rate in words per second, Allowed values are: (64, 128, 256, 512, 1024,8192**)**	ALLOWED_VALUES		
			“A717_channel”: A717 channel number. Valid values are (1 and 2).	SIZE 1, UNSIGNED, RANGE 1-2		
		“QAR_passcode”: When QAR are compressed into ZIP format, if the QAR_passcode is supplied, it will be encrypted using this passcode.		64, TEXT		
		“QARbiphaseformat”: If true, QAR is formatted as Harvard bi-phase encoding. If false, QAR Is formatted as (DITS) bi-polar encoding. Allowed values are ("true" or "false").		Boolean		
		“QAR_file_name”: Naming convention to be followed for QAR files. Starts with A717 and ends with ADG. Variable fields are same as A429. Example: "A717_<Tail>_<FlightLeg>_<DateTime>_QAR_<ICAO>_ADG.zip"		64, TEXT		
		“QAR _fram es_w ord”	“**QAR_frames**”: Comma separated list of subframes for recording.	SIZE 4, UNSIGNED, RANGE 1-4		
			“**QAR_word**”: Comma separated list of words to be recorded.	SIZE 4, UNSIGNED, RANGE 1-8192		
A664_AFDX_rec_en able	Boolea n	A664 bus recording enable flag ("true" or "false").		Boolean		
A664_Afdx_rec_filt er	Array of JSON	“VLID”: Afdx virtual link identification number		5, UNSIGNED, RANGE 1-65535		
		“SubID”: Afdx sub virtual link identification number or message id		2, UNSIGNED, RANGE 1-99		
		“AfdxPort”: Afdx port number		5, UNSIGNED, RANGE 1-65535		
		“IPSrc”: Source IP address		64, IPADDRESS		
		“IpDst”: Destination IP address		64, IPADsDRESS		
ascb_data_bus_por t	numbe r	Port number for ascb bus recording		5, UNSIGNED, RANGE 1-65535		
ascb_over_eth_port	numbe r	Port number for ascb over the ethernet bus recording		5, UNSIGNED, RANGE 1-65535		
ttgbe_over_eth_po rt	numbe r	Port number for ttgbe bus recording		5, UNSIGNED, RANGE 1-65535		
cmc_data_port	numbe r	Port number for cmc bus recording		5, UNSIGNED, RANGE 1-65535		
max_file_size	numbe r	Maximum file size in MB		3, UNSIGNED, RANGE 1 - 999		
max_file_time	numbe r	Maximum file time in minutes		2, UNSIGNED, RANGE 0 - 59		
recordings_retentio n_period	numbe r	Number of days to retain the QAR log files and A429 channel recordings after creation if not successfully uploaded to the ground server		3, UNSIGNED, RANGE 1-365		

ID : SSS_4213

1AWObjectType : Description

Refer below example for the additional fields of the aircraftSetting.txt file and fleetsetting.txt file

Example:


```
{
	"ascb_data_bus_recording": true,
	"ascb_over_eth_recording": true,
	"ttgbe_over_eth_recording": false,
	"cmc_data_recording": true,
	"A429_channel_info": [{
			"a429_channel_rec_enable": true,
			"A429_channel": 1,
			"A429_channel_name": "GAMA",
			"A429_channel_iface": "Core A429",
			"A429_hi_rate": true,
			"A429_channel_tx": false
		},
		{
			"a429_channel_rec_enable": true,
			"A429_channel": 2,
			"A429_channel_name": "IRU",
			"A429_hi_rate": false,
			"A429_channel_tx": false
		}
	],
	"A429_record_filter": {
		"A429_file_name": "A429_N1234P_Taxi_20170130_034533_GAMA_KPHX_ADG.zip",
		"A429_passcode": "asdfghjklzxcvbnmqwertyuiop",
		"record_filter": [{
				"A429_channel": 1,
				"A429_channel_iface": "Core A429",
```

```
                "A429_label": [277, 425],
                "A429_record_rate": 1,
                "A429_parity": false,
                "A429_ssm": 1
            },
            {
                "A429_channel": 2,
                "A429_label": [310, 311],
                "A429_record_rate": 100,
                "A429_parity": false,
                "A429_ssm": 1
            }
        ]
},
"A717_QAR_rec": {
        "A717_QAR_rec": true,
        "A717_channel_info": [{
                "A717_framerate": 64,
                "A717_channel:1
                },
                {
                "A717_framerate": 128,
                "A717_channel:2
                }
            ],
        "QARbiphaseformat": false,
        "QAR_passcode": "xyz12345678abc",
        "QAR_file_name": "A717_N1234P_Taxi_20170130_034533_QAR_KPHX_ADG.zip",
        "QAR_frames_word": [{
```

```
                        "QAR_frames": [1, 3],
                        "QAR_word": [33, 44]
                    },
                    {
                        "QAR_frames": [2, 4],
                        "QAR_word": [10, 14, 32]
                    }
                ]
            },
            "A664_Afdx_rec_enable": true,
            "A664_afdx_rec_filter": [{
                "VLID": 4701,
                "SubID": [2011, 2001],
                "AfdxPort": 16462,
                "IPSrc": "10.2.38.1",
                "IpDst": "224.224.27.197"
            }],
            "ascb_data_bus_port": 4001,
            "ascb_over_eth_port": 4002,
            "ttgbe_over_eth_port": "",
            "cmc_data_port": 4003,
            "max_file_size": 500,
            "max_file_time": 30,
            "recordings_retention_period ": 60
}
```


ID : SSS_4662

1AWObjectType : Requirement

The aircraftSetting.txt file and fleetsetting.txt file **shall** contain the following additional fields for Data Bus Streaming:

TAG	Format	Description	Default Value	Functional Area

a429_stream_Info	JSON	“A429_bus_stream_enable”: A429 bus streaming enable flag ("true" or "false"). If false, then the no attempts to stream any of A429 data bus.
		“A429_channel”: A429 bus to be stream, valid values are (1, 2, 3, 4).
		“A429_stream_app_limit”: Number of simultaneous data streaming app users.
		“A429_stream_interval”: Interval in second at which ADG-400 should stream the data.
			{
			"databusstreaming": {
			"a429_stream_info": {
			"A429_bus_stream_enable": true,
			"A429_channel": 1,
			"A429_stream_app_limit": 8,
			"A429_stream_interval": 1
			},
			"ascb_stream_info": {
			"ascb_bus_stream_enable": false
			},
			"ethernet_stream_info": {
			"ethernet_bus_stream_enable": false
			}
			}
			}
				Data Bus Streaming

ascb_stream_Info	JSON	“ascb_bus_stream_enable”: ASCB bus stream enable flag ("true" or "false"). If false, then the no attempts to stream ASCB data bus. “ascb_stream_app_limit”: Number of simultaneous data streaming app users. “ascb_stream_interval”: Interval in second at which ADG-400 should stream the data.		
ethernet_stream_Info	JSON	“ethernet_bus_stream_enable”: Ethernet bus stream enable flag ("true" or "false"). If false, then the no attempts to stream Ethernet data bus. “ethernet_stream_app_limit”: Number of simultaneous data streaming app users. “ethernet_stream_interval”: Interval in second at which ADG-400 should stream the data.		

ID : SSS_4663

1AWObjectType : Requirement

Example:

{

"dataBusStreaming_info": {

"a429_stream_info": {

"a429_bus_stream_enable": true,

"a429_channel": 1,

"a429_stream_app_limit": 8,

"a429_stream_interval": 1

},

"ascb_stream_info": {

"ascb_bus_stream_enable": false

"ascb_stream_app_limit":8,

```
                "ascb_stream_interval": 1,
                },
        "ethernet_stream_info": {
                "ethernet_bus_stream_enable": false,
                "ethernet_stream_app_limit":8,
                "ethernet_stream_interval": 1,

                }
        }
}
```

ID : SSS_4216

1AWObjectType : Requirement

The airportsetting.txt file **shall** be formatted in JSON format and contains the following fields

TAG	Format	Character Maximum Length, Size, Type, Allowed Values	Description	Default	Functional Area
airport_code	string	AIRPORT CODE	ICAO	No Factory Default	Network Manager
wifi_channel	number	WIFICHANNEL	Identifies which Channel to use for the Hot Spot when at this location. If value is blank or field is not included, then the CountryConSetting.txt channel is used.		Network Manager
cellular_mode_enable	boolean	N/A	Toggle cellular on and off. Allowed values are (“true” or “false”)		Network Manager
access_mode_enable	boolean	N/A	Toggle access point on and off. Allowed values are (“true” or “false”)		Network Manager
wifi_mode_enable	boolean	N/A	Toggle wi-fi on and off. Allowed values are (“true” or “false”)		Network Manager
network_priority	Array of String (optional)	SIZE 3, ALLOWED_VALUES	Ordered list of networks to use to connect to ground station when at this location. Allowed values are Wi-Fi, Cellular1 and Cellular2. If value is blank or field is not included, then the CountryConSetting.txt priority is used		Network Manager

Example:


```
{
"airportSetting": [
{
    "airport_code": "KSDF",
    "wifi_channels": 6,
    "cellular_mode_enable": true,
    "access_mode_enable": true,
    "wifi_mode_enable": true,
    "network_priority": [
```

"cellular1",

"cellular2",

"wifi"

]

}

]

}

ID : SSS_4215

1AWObjectType : Requirement

The airportMaster.txt file **shall** be formatted in JSON format and contains the following fields. This file gives the airport and country code for the given GPS co-ordinates.

TAG	Format	Character Maximum Length, Type, Range, Allowed Values	Description	Default	Functional Area
Latitude	number	16, FRACTIONAL, RANGE -90 to +90	Airport Latitude in degrees	No Factory Default UCF	Network Manager
Longitude	number	16, FRACTIONAL, RANGE -180 to +180	Airport Longitude in degrees	No Factory Default UCF	Network Manager
airport_code	String	AIRPORT CODE	ICAO	No Factory Default UCF	Network Manager
country_code	String	COUNTRYCODE	ISO 3166-1 alpha-3 Country Code	No Factory Default UCF	Network Manager
wifi_max_channel	number	WIFICHANNEL	Max allowed wifi channel (e.g. if value is 11, allowed channels are 1-11)	No Factory Default UCF	Network Manager
wifi_max_power	number	ALLOWED_VALUES	Max legal wifi power allowed at this location in dbm Allowed values are (10, 20 or 30).	No Factory Default UCF	Network Manager

Example


```
{
    "airportMaster": [
    {
        "latitude": 33.4342995,
        "longitude": -112.012001,
        "airport_code": "KPHX",
        "country_code": "USA",
        "wifi_max_channel": 11,
```

```
        "wifi_max_power": 30
    },
    {
        "latitude": 33.9425011,
        "longitude": -118.4078871,
        "airport_code": "KLAX",
        "country_code": 310,
        "wifi_max_channel": 13,
        "wifi_max_power": 30
    }
  ]
}
```


ID : SSS_5266

1AWObjectType : Requirement

The countryMapping.txt file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum Length, Type, Range, Allowed Values	Description	Default	Functional Area
Iso_code	String	COUNTRYCODE	ISO 3166-1 aplha-3 Country Code	No factory Default UCF	Network Manager
mcc_code	number	Three Digit Integer	MCC code of the country	No factory Default UCF	Network Manager

Example:


```
{
"countryMapping":[
        {
        "iso_code":"USA",
        "mcc_code":310
        },
        {
```

"iso_code":"USA",

"mcc_code":311

}

]

}

ID : SSS_4211

1AWObjectType : Requirement

The debug.txt configuration file **shall** be formatted in JSON format and contains the following fields:

TAG	Format	Character Maximum Length	Description	Default	Functional Area
a615_extended_logging	Boolean	N/A	If true, then all protocol command/responses are logged to the Activity log. Note, when this flag is set, data loading may be slower. Allowed values are "true" or "false"	No Factory Default UCF	615A Target Loading

Example:

{

" a615_extended_logging ": false

}

ID : SSS_4210

1AWObjectType : Requirement

The lanTestSetting.txt file **shall** be formatted in JSON format and contains an array of the following fields

TAG	Format	Description	Maximum Character Length, Type	Default	Functional Area
lantest_enable	Boolean	Flag for controlling the lan test.	N/A	No Factory Default UCF	FTP Server
hostdetails	Array of JSON	"host_name": Hostname to ping,	64, TEXT	No Factory Default UCF	FTP Server
		"ip_address": IP address of the host	64, IPADDRESS		

Example:

{

```
"lantestsettings": [
        {
        "lantest_enable": true,
           {
           “hostdetails”: [
                  {
                   "hostname": “printer”,
                   "ip_address": “192.168.200.1”,
                  },
                   {
                    "hostname": “Radar”,
                    "ip_address": “192.168.210.2”,
                   },
                   ]
              }
        }
  ]
 }
```


Note: lantestsettings.txt file defines the host name and the associated IP addresses. It’s also allows the user to control the LAN test.

ID : SSS_4221

1AWObjectType : Description

The ADG-400 UCF file contains the following certificates/keys.

Cert/Keys name	Character Maximum Length	Description	Default	Functional Area
config/UCFGroundEncryptKey.private	N/A	UCFGroundEncryptKey.private is a private key (RSA 2048) provided by Honeywell or Third Party Certificate provider used to decrypt the data in the UCF file.		Configuration Manager
config/UCFGroundEncryptKey.passphrase	N/A	UCFGroundEncryptKey.passphrase is a text file having the passphrase for the UCFGroundEncryptKey.private		Configuration Manager
config/UCFSigningCA.CER	N/A	UCFSigningCA.CER is a CA (x509v3) provided by Honeywell or Third Party Certificate provider used to validate the Signing Public certificate which is received along with the signature file and data in the UCF file.		Configuration Manager
config/UCFSigningCA.CRL	N/A	UCFSigningCA.CRL is a CRL (x509v2) provided by Honeywell or Third Party Certificate provider for the certificate provided by UCFSigningCA.CER		Configuration Manager
sslserver/standardServerSSLKey.p12	N/A	standardServerSSLKey.CER is a public certificate (x509v3) provided by HON for the access point https communication protocol between Client application and SDG-400.		Security, Remote Terminal/Apps connection Management
sslserver/standardClientSSLCA.CER	N/A	StandardClientSSLCA.CER is a CA(x509v3) provided by HON. Used to validate the client certificate for https 2 way SSL handshake between Client application and SDG-400.		Security, Remote Terminal/Apps connection Management
sslserver / standardClientSSLCA.CRL	N/A	standardClientSSLCA.CRL is a CRL (x509v2) provided by HON PKI Provider/ Operator PKI Provider for the certificate provided by standardClientSSLCA.CER		
sslclient/standardClientSSLKey.P12	N/A	standardClientSSLKey.P12 is a public private key pair provided by HON PKI Provider/ Operator PKI Provider for the access point https communication protocol between ground server and SDG-400.		Data Transfer, Security
sslclient/standardServerSSLCA.CER	N/A	standardServerSSLCA.CER is a CA(x509v3) provided by HON PKI Provider/ Operator PKI Provider. Used to validate the server certificate for https 2 way SSL handshake between ground server and SDG-400.		Data Transfer, Security
wificlient/wifiHotSpotCA.CER	N/A	wifiHotspotCA.CER is a CA(x509v3) provided by Operator PKI Provider. Used to validate the ground wifi hotspot server certificate.		Network Manager
wificlient/wifiHotSportKey.private	N/A	wifiHotspotKey.private is a private key (RSA 2048) used to connect ground Wi-Fi.		Network Manager
radiusserver/radiusServerKey.p12	N/A	raduisServerKey.p12 is provided by HON PKI Provider/Operator PKI Provider which is used by radius server for WAP.		Network Manager
radiusserver/wifiClientCA.CER	N/A	wifiClientCA.CER is a CA(x509v3) provided by HON PKI Provider/ Operator PKI Provider. Used to validate the mobile device client certificate for WPA2/AES/Enterprise communication between mobile device and OBS.		Network Manager
radiusserver/wifiClientCA.CRL	N/A	wifiClientCA.CRL is a CRL (x509v2) provided by HON PKI Provider/Operator PKI Provider for the certificate provided by wifiClientCA.CER		Network Manager

ID : SSS_5206

1AWObjectType : Requirement

The aircraftsetting.txt file **shall** contain the following fields for Data off-boarding:

TAG	Format	Character Maximum length, type	Description	Default	Functional Area

dataoffboarding_info	Boolean
	String
	String
			"automatic_off_boarding": Whether to offboard the data automatically when the aircraft is in air.Automatic off boarding enable flag ("true" or "false"). If false, then no attempts to automatic off-boarding will be made.
			"offboarding_source": Source path from where the recorded flight data in ADG-400 is stored and is picked for off-boarding Ex: WDLS
			"offboarding_destination": Destination path where the off-boarded data is stored either in SD card or USB drive if inserted.
				{
				"dataoffboarding": {
				"automatic_off_boarding_enable": true,
				"offboarding_source": DATA,
				"offboarding_destination": SD
				}
				}
					Data off-boarding
					Data off-boarding
					Data off-boarding

Example:

"dataOffboardingInfo":{

"automatic_off_boarding": true,

"offboarding_source":” Data”,

"offboarding_destination": "SD"

}

Note: DATA: This folder contain data like QAR, Ethernet, etc.

ID : SSS_145

1AWObjectType : Title

3.3.6.1.2 ADG-400 Configuration File Security

ID : SSS_166

1AWObjectType : Requirement

The ADG-400 **shall** only accept a new software load, ACF, or HCF if received on the wired Avionics LAN.

ID : SSS_167

1AWObjectType : Requirement

The ADG-400 **shall** reject a load of the Application SW, ACF, or HCF from the ADG-400 Data Transfer Function.

ID : SSS_168

1AWObjectType : Description

The purpose of the above requirements is to prevent Wireless or self loading of these items. These items must be updated by an aircraft maintenance technician.

ID : SSS_153

1AWObjectType : Requirement

The HCF and UCF **shall** be signed and encrypted.

ID : SSS_154

1AWObjectType : Requirement

The ACF **shall** be signed and not encrypted.

ID : SSS_2407

1AWObjectType : Requirement

The ADG-400 **shall** verify a signed ACF using the current HCF Honeywell Public Signing Certificate.

ID : SSS_155

1AWObjectType : Requirement

The ADG-400 **shall** use only Honeywell Signed Certificates to verify the HCF file signature.

ID : SSS_164

1AWObjectType : Requirement

The ADG-400 **shall** store default decryption key with the operating system software.

ID : SSS_165

1AWObjectType : Description

The decryption (public) key will be included in the operational software, and changes to the key will require a software update of the ADG-400, The web service that generates the key pair, encrypts data and signs files will be responsible for maintaining the security of the encryption (private) key.

Note: The decryption key will be provided separately from this specification to maintain the security of the key.

ID : SSS_2438

1AWObjectType : Requirement

The UCF Factory Default Private/Public Keys **shall** becomes unavailable to ADG-400 once user loadable keys are loaded unless the ADG-400 is put back into factory default.

ID : SSS_2388

1AWObjectType : Title

3.3.6.1.3 ADG-400 Configuration File Installation

ID : SSS_2441

1AWObjectType : Description

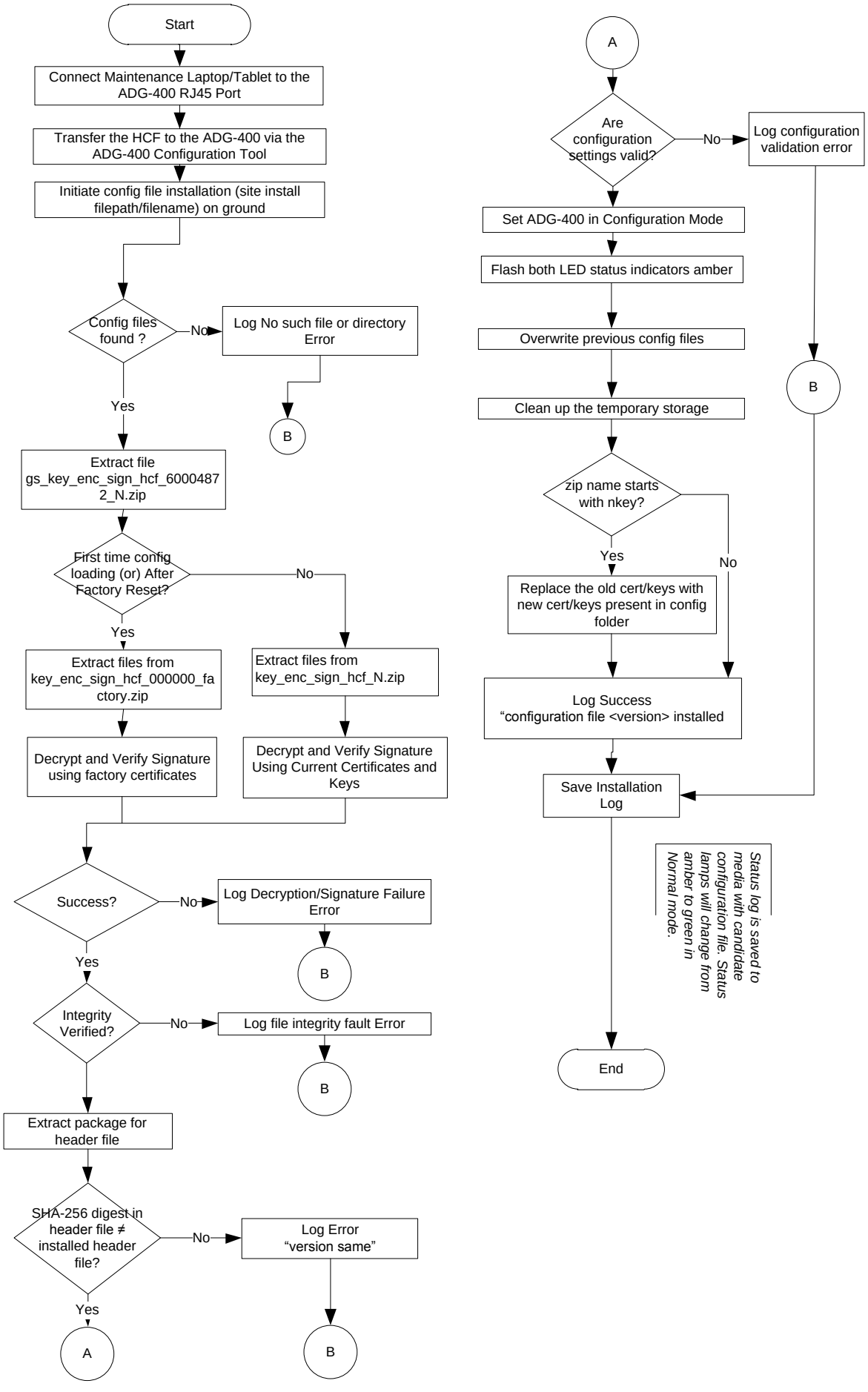
Installing the configuration on the ADG-400 will be initiated when configuration files are present at the root directory and the ADG-400 system receives FTP command site install <filename>. The ADG-400 will decrypt the file to verify the integrity, check the digest and perform the configuration settings validation before entering in configuration mode for storing the encrypted configuration file to a secure memory location that cannot be accessed by external clients. The encrypted configuration file will be retained until over written by installing a new configuration file.

During configuration file installation, all the files should be extracted in temporary storage. The ADG-400 will cleanup the temporary store after successful installation of the files.

ID : SSS_2442

1AWObjectType : Description

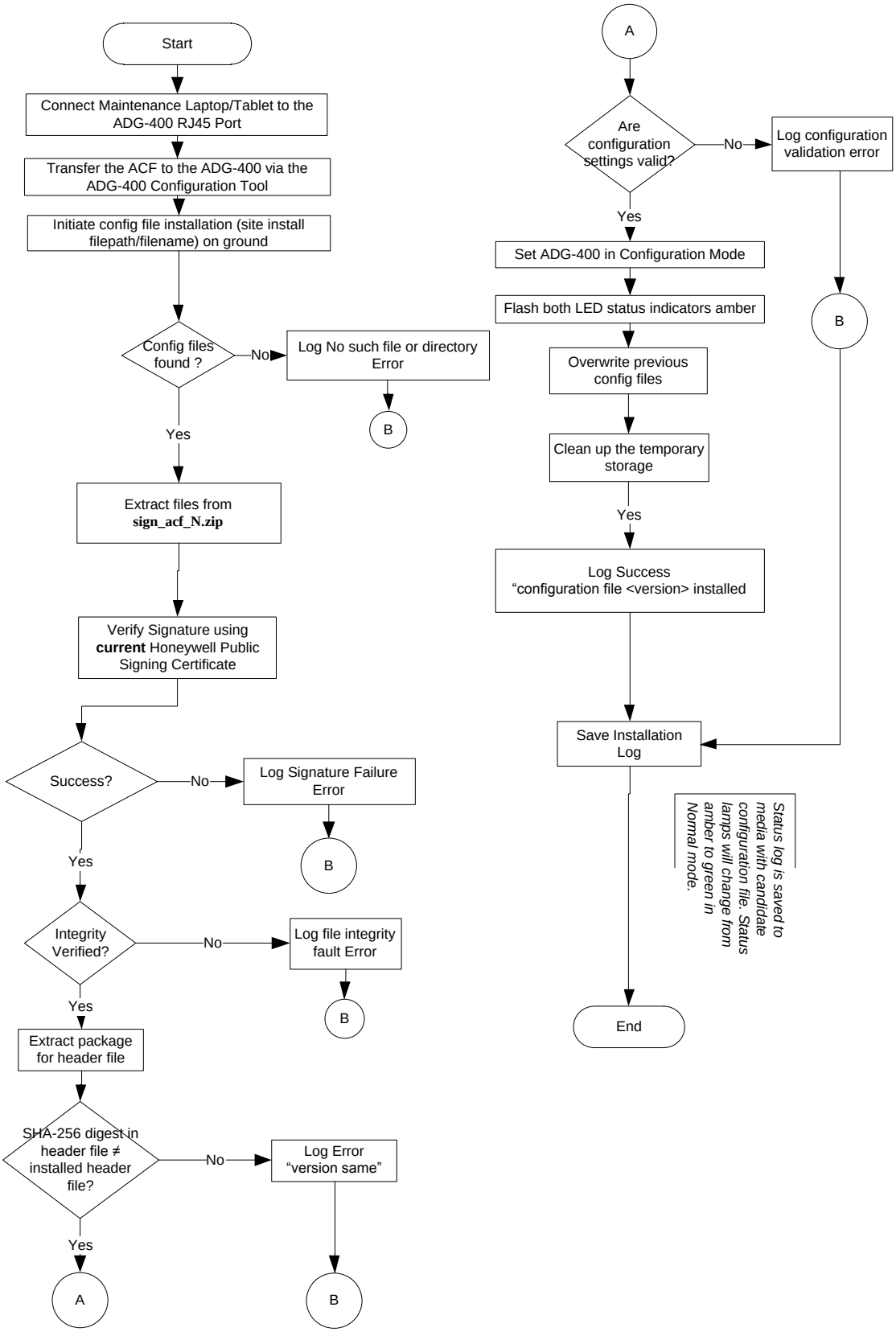
Diagram below describes the HCF configuration file installation.


Start
Connect Maintenance Laptop/Tablet to the ADG-400 RJ45 Port
Transfer the HCF to the ADG-400 via the ADG-400 Configuration Tool
Initiate config file installation (site install filepath/filename) on ground
Config files found ?
No
Log No such file or directory Error
B
Yes
Extract file gs_key_enc_sign_hcf_60004872_N.zip
First time config loading (or) After Factory Reset?
No
Yes
Extract files from key_enc_sign_hcf_000000_factory.zip
Extract files from key_enc_sign_hcf_N.zip
Decrypt and Verify Signature using factory certificates
Decrypt and Verify Signature Using Current Certificates and Keys
Success?
No
Log Decryption/Signature Failure Error
B
Yes
Integrity Verified?
No
Log file integrity fault Error
B
Extract package for header file
SHA-256 digest in header file ≠ installed header file?
No
Log Error "version same"
B
Yes
A
A
Are configuration settings valid?
No
Log configuration validation error
B
Set ADG-400 in Configuration Mode
Flash both LED status indicators amber
Overwrite previous config files
Clean up the temporary storage
zip name starts with nkey?
Yes
No
Replace the old cert/keys with new cert/keys present in config folder
Log Success "configuration file <version> installed
Save Installation Log
End
Status log is saved to media with candidate configuration file. Status lamps will change from amber to green in Normal mode.

ID : SSS_2443

1AWObjectType : Description

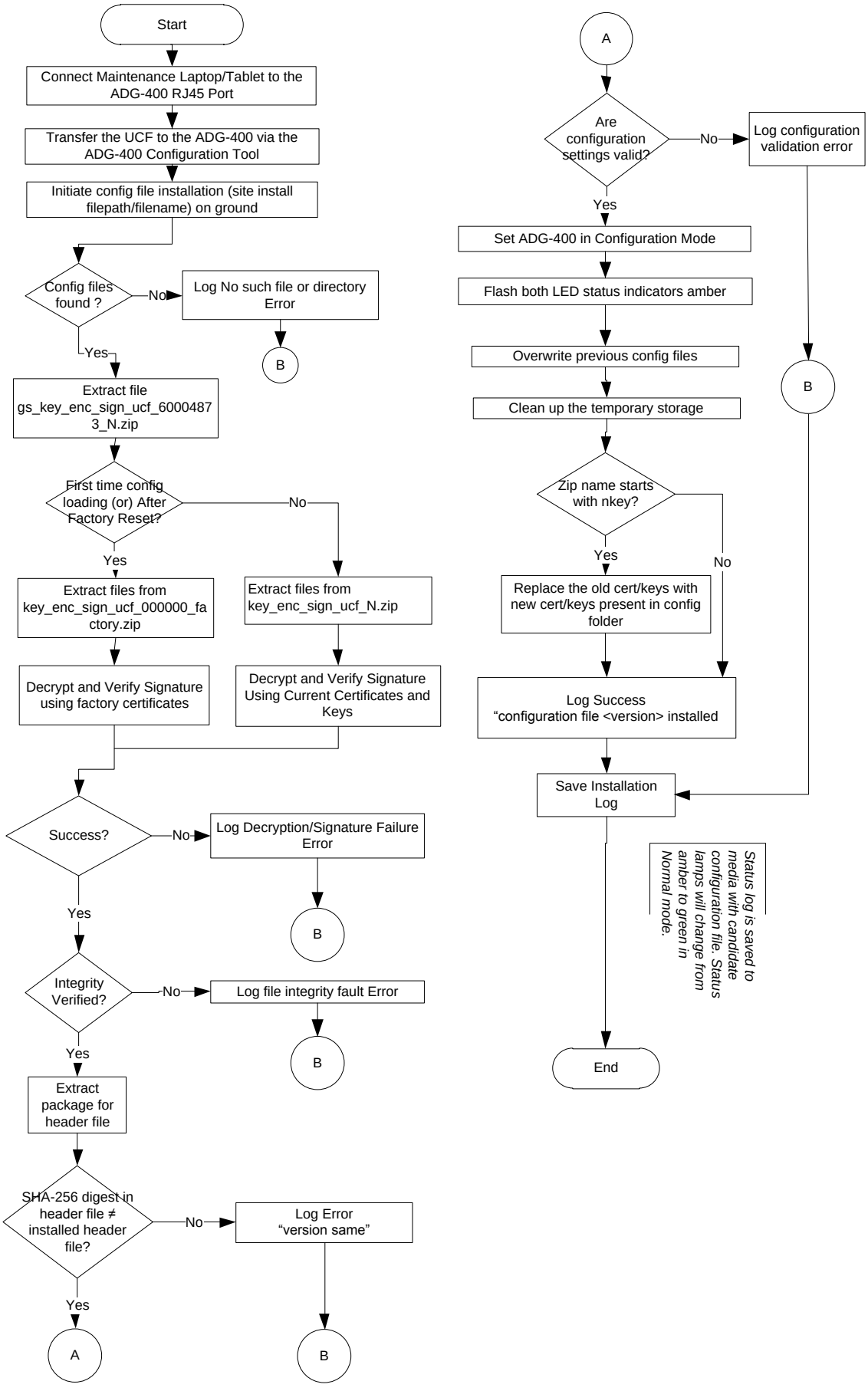
Diagram below describes the ACF configuration file installation.


Start
Connect Maintenance Laptop/Tablet to the ADG-400 RJ45 Port
Transfer the ACF to the ADG-400 via the ADG-400 Configuration Tool
Initiate config file installation (site install filepath/filename) on ground
Config files found ?
No
Log No such file or directory Error
B
Yes
Extract files from **sign_acf_N.zip**
Verify Signature using **current** Honeywell Public Signing Certificate
Success?
No
Log Signature Failure Error
B
Yes
Integrity Verified?
No
Log file integrity fault Error
B
Yes
Extract package for header file
SHA-256 digest in header file ≠ installed header file?
No
Log Error "version same"
B
Yes
A
A
Are configuration settings valid?
No
Log configuration validation error
B
Yes
Set ADG-400 in Configuration Mode
Flash both LED status indicators amber
Overwrite previous config files
Clean up the temporary storage
Yes
Log Success "configuration file <version> installed
Save Installation Log
End
Status log is saved to media with candidate configuration file. Status lamps will change from amber to green in Normal mode.

ID : SSS_2444

1AWObjectType : Description

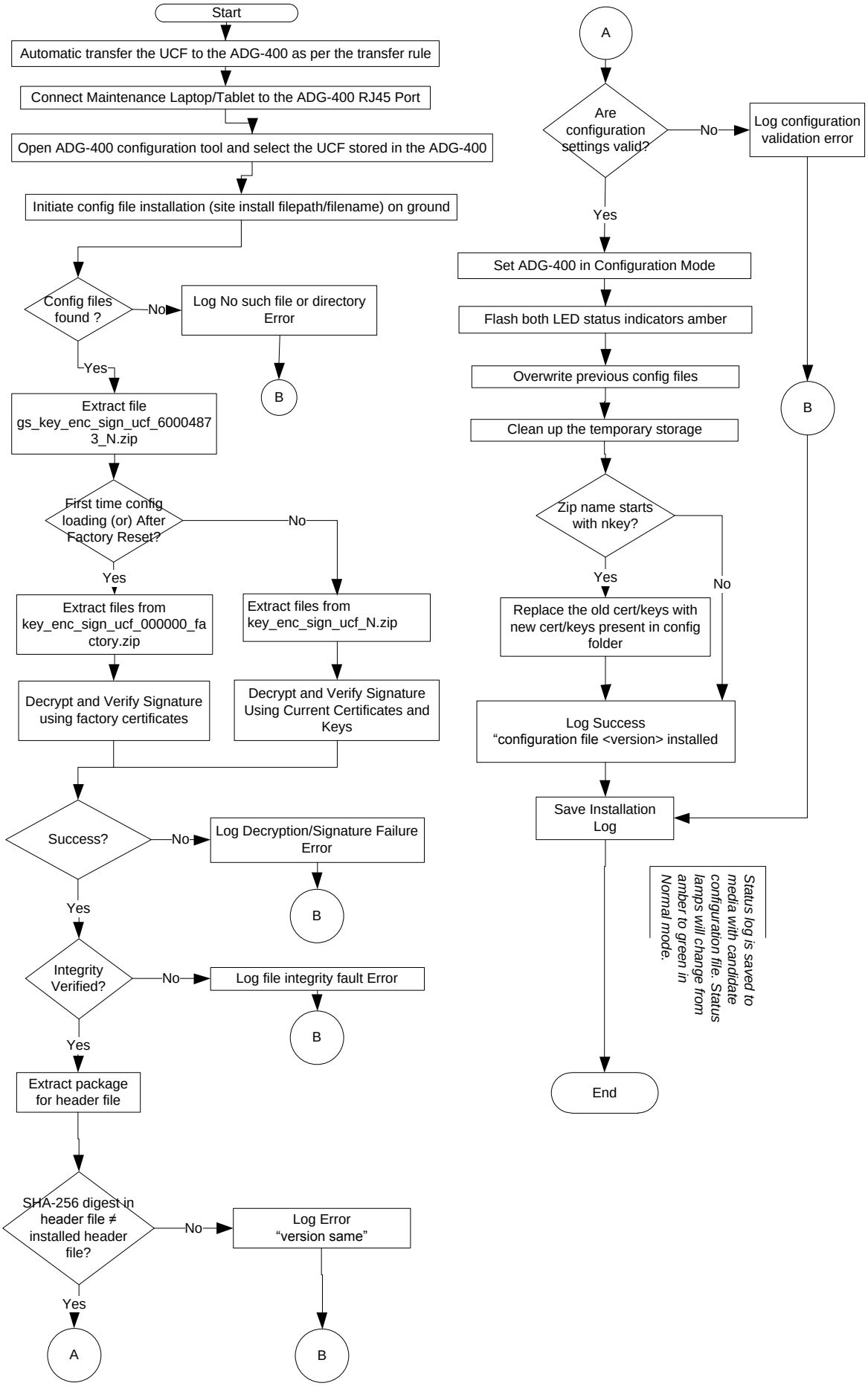
Diagram below describes the UCF configuration file installation.


Start
Connect Maintenance Laptop/Tablet to the ADG-400 RJ45 Port
Transfer the UCF to the ADG-400 via the ADG-400 Configuration Tool
Initiate config file installation (site install filepath/filename) on ground
Config files found ?
No
Log No such file or directory Error
B
Yes
Extract file gs_key_enc_sign_ucf_60004873_N.zip
First time config loading (or) After Factory Reset?
No
Yes
Extract files from key_enc_sign_ucf_000000_factory.zip
Extract files from key_enc_sign_ucf_N.zip
Decrypt and Verify Signature using factory certificates
Decrypt and Verify Signature Using Current Certificates and Keys
Success?
No
Log Decryption/Signature Failure Error
B
Yes
Integrity Verified?
No
Log file integrity fault Error
B
Yes
Extract package for header file
SHA-256 digest in header file ≠ installed header file?
No
Log Error "version same"
B
Yes
A
A
Are configuration settings valid?
No
Log configuration validation error
B
Yes
Set ADG-400 in Configuration Mode
Flash both LED status indicators amber
Overwrite previous config files
Clean up the temporary storage
Zip name starts with nkey?
Yes
No
Replace the old cert/keys with new cert/keys present in config folder
Log Success "configuration file <version> installed
Save Installation Log
Status log is saved to media with candidate configuration file. Status lamps will change from amber to green in Normal mode.
End

ID : SSS_2445

1AWObjectType : Description

Diagram below describes the UCF configuration file installation **(Automatic Transfer).**


Start
Automatic transfer the UCF to the ADG-400 as per the transfer rule
Connect Maintenance Laptop/Tablet to the ADG-400 RJ45 Port
Open ADG-400 configuration tool and select the UCF stored in the ADG-400
Initiate config file installation (site install filepath/filename) on ground
Config files found ?
No
Log No such file or directory Error
B
Yes
Extract file gs_key_enc_sign_ucf_60004873_N.zip
First time config loading (or) After Factory Reset?
No
Yes
Extract files from key_enc_sign_ucf_000000_factory.zip
Extract files from key_enc_sign_ucf_N.zip
Decrypt and Verify Signature using factory certificates
Decrypt and Verify Signature Using Current Certificates and Keys
Success?
No
Log Decryption/Signature Failure Error
B
Yes
Integrity Verified?
No
Log file integrity fault Error
B
Yes
Extract package for header file
SHA-256 digest in header file ≠ installed header file?
No
Log Error "version same"
B
Yes
A
A
Are configuration settings valid?
No
Log configuration validation error
B
Yes
Set ADG-400 in Configuration Mode
Flash both LED status indicators amber
Overwrite previous config files
Clean up the temporary storage
Zip name starts with nkey?
Yes
Replace the old cert/keys with new cert/keys present in config folder
No
Log Success "configuration file <version> installed
Save Installation Log
Status log is saved to media with candidate configuration file. Status lamps will change from amber to green in Normal mode.
End

ID : SSS_2355

1AWObjectType : Requirement

When all of the Checks as mentioned below are successful during HCF/UCF configuration installation, the ADG-400 **shall** transition to configuration mode if system is on ground and store the following configuration files on the SSD card or TPM memory (TBD):

Configuration Files:

if the box is new or box is factory reset:

- For HCF installation: key_enc_sign_hcf_000000_factory.zip,
- For UCF installation: key_enc_sign_ucf_000000_factory.zip

Otherwise,

- For HCF installation: key_enc_sign_hcf_N.zip,
- For UCF installation: key_enc_sign_ucf_N.zip

Check:

a). File Decryption

b). File Signature Verification

c). File Integrity

d). Version Check (not required if the box is new or box is factory reset)

e). Configuration Validation

Refer SSS_2442, SSS_2443, SSS_2445 for the installation steps.

ID : SSS_4829

1AWObjectType : Requirement

When all of the Checks as mentioned below are successful during ACF configuration installation, the ADG-400 **shall** transition to configuration mode if system is on ground and store the following configuration files on the SSD card or TPM memory (TBD):

Configuration Files:

sign_acf_N.zip

Check:

a). File Signature Verification

b). File Integrity

c). Version Check (not required if the box is new or box is factory reset)

d). Configuration Validation

Refer SSS_2444 for the installation steps..

ID : SSS_2389

1AWObjectType : Requirement

When in data load mode, the ADG-400 **shall** perform ACF/HCF configuration loading when initiated only from an dataloader application running on a maintenance laptop/tablet connected through hardwired to the RJ45 ethernet port of the ADG-400 using https communications.

ID : SSS_163

1AWObjectType : Requirement

The UCF **shall** be modifiable via the ADG-400 Wireless Connection, the ADG-400 Wired Connection, or the Data Transfer Function.

ID : SSS_4730

1AWObjectType : Requirement

During HCF configuration installation, the ADG-400 **shall** extract the gs_key_enc_sign_hcf_60004872_N.zip file to get files:

- key_enc_sign_hcf_N.zip,
- key_enc_sign_hcf_000000_factory.zip

ID : SSS_4729

1AWObjectType : Requirement

During UCF configuration installation, the ADG-400 **shall** extract the gs_key_enc_sign_hcf_60004873_N.zip file to get files:

- key_enc_sign_ucf_N.zip,
- key_enc_sign_ucf_000000_factory.zip

ID : SSS_175

1AWObjectType : Requirement

The ADG-400 **shall** install the sign_acf_N.zip only after installation of HCF.

ID : SSS_2406

1AWObjectType : Requirement

The ADG-400 **shall** perform the configuration installation in the NVM when initiated through the FTP command site install through ADG-400 Configuration Tool installed on a Maintenance/Tablet wired to Avionics LAN of the ADG-400.

Commentary: The UCF configuration file should be present on the Maintenance Laptop/Tablet. The UCF file can be download from SIGNS server or create using ADG-400 Configuration Tool. For initiating configuration installation, the user has to select the UCF file from the maintenance laptop/tablet or select the file present on ADG-400. The ADG-400 Data Transfer Function (DTF) can be used to stage the UCF configuration file on the ADG-400 storage.

The HCF/ACF files should be present on the Maintenance Laptop/Tablet for installation. These files can not be transfer using DTF or push through wireless.

ID : SSS_4672

1AWObjectType : Requirement

When configuration settings data validation fails during configuration file installation, the ADG-400 **shall** abort the configuration file installation and log the informational error message as defined under ADG-400 Configuration Data Distribution section.

ID : SSS_148

1AWObjectType : Requirement

The ADG-400 **shall** implement the new configuration settings at the next processor restart.

ID : SSS_149

1AWObjectType : Guidance

A power up restart, or a commanded reset can be used to apply the new configuration setting.

The dedicated persistent memory stores security sensitive files and must be protected from external read, write and delete activities.

The configuration file, and in particular the keys required for the RADIUS server, are stored encrypted and are only decrypted in RAM at startup.

ID : SSS_2390

1AWObjectType : Requirement

When in data load mode, the ADG-400 **shall** perform UCF configuration loading when initiated from an dataloader application:

- Running on a mobile device securely connected through wireless to the ADG-400 using https communications.

  OR

- Running on a maintenance laptop/tablet connected hardwired to the RJ45 ethernet port of the ADG-400 using https communications.

ID : SSS_2391

1AWObjectType : Description

Once ADG-400 in the Dataload Mode and the Dataloader Application has securely connected to the ADG-400, the application will send the selected load to the ADG-400 Target Loader function within the ADG-400 for loading or indicate the location of the load in the ADG-400 data store.

The tablet/device/maintenance laptop used for loading should have a certificate provided by SIGNS and have the Dataloader Application loaded.

ID : SSS_2392

1AWObjectType : Requirement

The ADG-400 **shall** store the configuration files in the /CONFIG folder of ADG-400 when ARINC 665 package is data loaded using a standard 615A data loader.

ID : SSS_4225

1AWObjectType : Description

Each configuration file has a header which contains a SHA-256 digest of all of the files excluding the header file in alphabetical order. This field is to fulfill the ADG-400 requirement to detect and prevent tampering with any parameters of the configuration files. This field is automatically calculated when the file is created or updated and before applying the below described Encryption and Signing. Any files added to the master zip file for supporting encryption or signing are not included in the digest calculation.

When a configuration file is loaded into the ADG-400, the ADG-400 will verify the digest stored in the candidate header file against the digest calculated.

ID : SSS_4828

1AWObjectType : Title

3.3.6.1.3.1 Decrypt and Verify Signature Using Current Certificates and Keys

ID : SSS_2465

1AWObjectType : Requirement

During ADG-400 HCF configuration installation or at initialization after power up, the ADG-400 **shall** perform the following steps to read a signed and encrypted HCF using the current set of certificates and keys.

Steps:

1. Extract files from key_enc_sign_hcf_N.zip.

2. Decrypt the symmetric key [enc_symmetric.key] with the current Honeywell Private Encryption Key [HCFGroundEncryptKey.private]. An Open SSL command-line example is:

   openssl smime -decrypt -binary -aes256 -in enc_symmetric.key -inform DER -out symmetric.key -inkey HCFGroundEncryptKey.private -passin file:HCFGroundEncryptKey.passphrase

3. Using the AES-256 algorithm, decrypt the enc_sign_hcf_N.zip file with the symmetric key [symmetric.key]. An Open SSL command-line example is: openssl enc -d -aes-256-cbc -md sha256 -in enc_sign_hcf_000001.zip -out sign_hcf_000001.zip -pass file:symmetric.key

4. Extract files from sign_hcf_000001.zip.

5. Verify the extracted public certificate with Public Signing CA with CRL openssl verify -crl_check -CAfile HCFSigningCA.CER -CRLfile HCFSigningCA.CRL HCFGroundSigningKey.CER

6. Verify the extension of the Certificate HCFGroundSigningKey.CER:

   X509v3 Key Usage:

   Digital Signature, Non Repudiation, Off-line CRL Signing

7. Verify the signature of the hash [hcf_000001.bin] using the current Honeywell Public Signing Certificate [HCFGroundSigningKey.CER]. An Open SSL command-line example is: openssl x509 -

pubkey -noout -in HCFGroundSigningKey.CER > pubkey.pem openssl dgst -sha256 -verify pubkey.pem -signature hcf_000001.bin hcf_000001.zip

ID : SSS_2459

1AWObjectType : Requirement

The ADG-400 **shall** perform the following steps to read a signed and encrypted UCF using the current set of certificates and keys.

Steps:

1. Extract files from key_enc_sign_ucf_000001.zip.

2. Decrypt the symmetric key [enc_symmetric.key] with the current Customer Private Encryption Key [UCFGroundEncryptKey.private]. An Open SSL command-line example is: openssl smime -decrypt -binary -aes256 -in enc_symmetric.key -inform DER -out symmetric.key -inkey UCFGroundEncryptKey.private -passin file:UCFGroundEncryptKey.passphrase

3. Using the AES-256 algorithm, decrypt the enc_sign_ucf_000001.zip file with the symmetric key [symmetric.key]. An Open SSL command-line example is: openssl enc -d -aes-256-cbc -md sha256 -in enc_sign_ucf_000001.zip -out sign_ucf_000001.zip -pass file:symmetric.key

4. Extract files from sign_ucf_000001.zip.

5. Verify the extracted public certificate with Public Signing CA with CRL openssl verify -crl_check -CAfile UCFSigningCA.CER -CRLfile UCFSigningCA.CRL UCFGroundSigningKey.CER

6. Verify the extension of the Certificate UCFGroundSigningKey.CER:

X509v3 Key Usage:

Digital Signature, Non-Repudiation

7. Verify the signature of the hash [ucf_000001.bin] using the current Customer Public Signing Certificate [UCFGroundSigningKey.CER]. An Open SSL command-line example is: openssl x509 -pubkey -noout -in UCFGroundSigningKey.CER > pubkey.pem openssl dgst -sha256 -verify pubkey.pem -signature ucf_000001.bin ucf_000001.zip

ID : SSS_2454

1AWObjectType : Requirement

The ADG-400 **shall** perform the following steps to read a signed ACF using the current set of certificate.

Steps:

1. Extract the loaded sign_acf_N.zip file to get acf_N.zip, acf_N.bin files.

2. Verify Signature using the current Honeywell Public Signing Certificate [HCFGroundSigningKey.CER]. An Open SSL command-line example is: openssl x509 -pubkey -noout -in HCFGroundSigningKey.CER > pubkey.pem openssl dgst -sha256 -verify pubkey.pem -signature acf_N.bin acf_N.zip

3. If signature verified successfully, load the acf_N.zip onto ADG-400.

ID : SSS_2393

1AWObjectType : Title

3.3.6.1.4 ADG-400 Configuration Data Distribution

ID : SSS_2394

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** perform the following configuration settings data validation tests on the HCF/ACF/UCF Configuration data relevant to it, for:

- Completeness (ADG-400 configuration data is present in the configuration data).
- Range validation (when the data is present, is the data within the defined range).

Note: The BITE section details the response to incorrect data.

ID : SSS_4669

1AWObjectType : Description

The ADG-400 overwrites the previous configuration files stored in NVM when configuration settings data validation tests succeeded during configuration file installation in the configuration. Refer SSS_2453 for the details.

ID : SSS_4750

1AWObjectType : Description

Refer SSS_4716 for the details on the configuration parameter format, range, type, JSON contents and others customized data type used throughout the configuration validation requirements.

Any text file is considered valid JSON file if it does not have any JSON parse error during validation. Refer SSS_2387 for the recommended syntax and data type in JSON format.

ID : SSS_4732

1AWObjectType : Requirement

Detected during HCF/UCF/ACF configuration parameter validation, by the ADG-400:

- JSON Parse Error
- Range Error
- Format Error
- Out of Bound Error

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF7 CONFIGURATION MANAGEMENT

L3: 0x01 HCF Parse Error

0x02 UCF Parse Error

0x03 ACF Parse Error

0x04 HCF Format Error

0x05 UCF Format Error

0x06 ACF Format Error

0x07 HCF Range Error

0x08 UCF Range Error

0x09 ACF Range Error

0x0A HCF Out of Bound Error

0x0B UCF Out of Bound Error

0x0C ACF Out of Bound Error

L4.Type: 01b EVENT

L4.Action: 0b N/A

L4.Level: 00b INFORMATION

L4.Hist: 001b OPERATIONAL

L4.Detect: 000b CONTINUOUS

- Additional Text: one or more parameters, in the following form:
  - <parameterlist> = <parameter> [';' <parameter>]
    - For JSON Parse Error, <parameter> = file name having JSON parse error.
    - For others configuration parameter error related, <parameter> = <filename> ';' <configuration> [';'<configuration>]
    - <filename> = name of the file having configuration parameter error.

    <configuration> = name of the configuration parameter.

Detection Action: RECORD

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_2446

1AWObjectType : Requirement

JSON Parse Error events within the ADG-400 configuration file **shall** be detected when validating JSON file during ADG-400 configuration installation.

ID : SSS_4739

1AWObjectType : Requirement

Format Error events within the ADG-400 configuration file **shall** be detected during ADG-400 configuration installation, in the valid JSON file, by the ADG-400 when one or more configuration parameters do not have a valid format.

ID : SSS_4738

1AWObjectType : Requirement

Range Error events within the ADG-400 configuration file **shall** be detected during ADG-400 configuration installation, in the valid JSON file, by the ADG-400 when one or more of the configuration parameter value is not in the set or range of allowed values.

ID : SSS_4740

1AWObjectType : Requirement

Out of Bound Error events within the ADG-400 configuration file **shall** be detected during ADG-400 configuration installation, in the valid JSON file, by the ADG-400, for one or more configuration parameters when:

- String size exceeds the defined limit.
- Array size exceeds the defined limit.

ID : SSS_4748

1AWObjectType : Requirement

Detected during HCF/UCF/ACF configuration parameter validation, by the ADG-400:

- Configuration parameters missing error

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF7 CONFIGURATION MANAGEMENT

L3: 0x 0D HCF CONFIGURATION PARAMETERS MISSING

0x 0E UCF CONFIGURATION PARAMETERS MISSING

0x 0f ACF CONFIGURATION PARAMETERS MISSING

L4.Type: 01b EVENT

L4.Action: 0b N/A

L4.Level: 00b INFORMATION

L4.Hist: 001b OPERATIONAL

L4.Detect: 000b CONTINUOUS

Additional Text: list of configuration parameters missing in the below format

- <parameterlist> = <parameter> [';' <parameter>]
- <parameter> = <filename> ';' <parametername> [';'< parametername>]
- <filename> = name of the file having configuration parameters missing error.

<parametername> = name of the missing configuration parameter.

Detection Action: RECORD

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_4749

1AWObjectType : Requirement

Configuration parameters missing events within the ADG-400 configuration file **shall** be detected during ADG-400 configuration installation, in the valid JSON file, by the ADG-400 for one or more configuration parameters when:

- Parameter TAG mismatch, OR
- Parameter TAG missing, OR
- Parameter value is empty or blank

ID : SSS_4311

1AWObjectType : Title

3.3.6.1.4.1 ADG-400 Configuration Validity Check

ID : SSS_4668

1AWObjectType : Title

3.3.6.1.4.1.1 Configuration Validation Check for HCF parameters.

ID : SSS_4312

1AWObjectType : Not a Requirement

This section defines the requirements for validating the configuration setttings in each of the Honeywell configuration file (HCF).

ID : SSS_4313

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the HCFheader.txt file as define in the SSS_4180 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4318

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the firewall.txt file as define in the SSS_4181 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4314

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the whiteList.txt file as define in the SSS_4771 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4317

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the authorizationList.txt file as define in the SSS_4183 against the valid JSON Format, TAG, Character Maximum Length, Type and Allowed Values.

ID : SSS_4316

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the lans.txt file as define in the SSS_4185 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4670

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the hardwaremode.txt file as define in the SSS_4189 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4671

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the softwareconfig.txt file as define in the SSS_4190 against the valid JSON Format, TAG, Character Maximum Length, Type and Array Size.

ID : SSS_4673

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the nvmpartitionscale.txt file as define in the SSS_4193 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4675

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the dataloadinginfo.txt file as define in the SSS_4682 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4705

1AWObjectType : Title

3.3.6.1.4.1.2 Configuration Validation Check for ACF parameters.

ID : SSS_4755

1AWObjectType : Description

This section defines the requirements for validating the configuration setttings in each of the Aircraft configuration file (ACF).

ID : SSS_4756

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the ACFheader.txt file as define in the SSS_4196 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4757

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the ACF aircraftSetting.txt file as define in the SSS_4197 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4704

1AWObjectType : Title

3.3.6.1.4.1.3 Configuration Validation Check for UCF parameters.

ID : SSS_4706

1AWObjectType : Description

This section defines the requirements for validating the configuration setttings in each of the UCF configuration file.

ID : SSS_4717

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the UCFheader.txt file as define in the SSS_4199 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4718

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the cellularSetting.txt file as define in the SSS_4200 against the valid JSON Format, TAG, Character Maximum Length, Type and Allowed Values.

ID : SSS_4719

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the countrySetting.txt file as define in the SSS_4201 against the valid JSON Format, TAG, Array Size, Type and Allowed Values.

ID : SSS_4725

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the wlanSetting.txt file as define in the SSS_4203 against the valid JSON Format, TAG, Character Maximum Length, Type.

ID : SSS_4724

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the transferSetting.txt file as define in the SSS_4205 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4723

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the apModeSetting.txt file as define in the SSS_4208 against the valid JSON Format, TAG, Character Maximum Length and Type.

ID : SSS_4722

1AWObjectType : Requirement

During configuration installation, the ADG-400 shall validate each of the configuration parameters in the queue.txt file as define in the SSS_4217 against the valid JSON Format, TAG, Character Maximum Length, Type, Range and Allowed Values.

ID : SSS_4721

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the aircraftSetting.txt file as define in the SSS_4219 against the valid JSON Format, TAG, Character Maximum Length, Allowed Values and Type.

ID : SSS_4720

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the airportSetting.txt file as define in the SSS_4216 against the valid JSON Format, TAG, Character Maximum Length, Array Size, Type and Allowed Values.

ID : SSS_4728

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the airportMaster.txt file as define in the SSS_4215 against the valid JSON Format, TAG, Character Maximum Length, Range, Type and Allowed Values.

ID : SSS_4727

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the fleetsetting.txt file as define in the SSS_4212 against the valid JSON Format, TAG, Character Maximum Length, Range, Type and Allowed Values.

ID : SSS_4726

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the debug.txt file as define in the SSS_4211 against the valid JSON Format, TAG and Type.

ID : SSS_4731

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the lanTestSetting.txt file as define in the SSS_4210 against the valid JSON Format, TAG, Character Maximum Length, and Type.

ID : SSS_4188

1AWObjectType : Requirement

During configuration installation, the ADG-400 **shall** validate each of the configuration parameters in the httpsServerSetting.txt file as define in the SSS_4206 against the valid JSON Format, TAG, Character Maximum Length, and Type.

ID : SSS_144

1AWObjectType : Title

3.3.6.1.5 ADG-400 Configuration Settings After Power Up

ID : SSS_2439

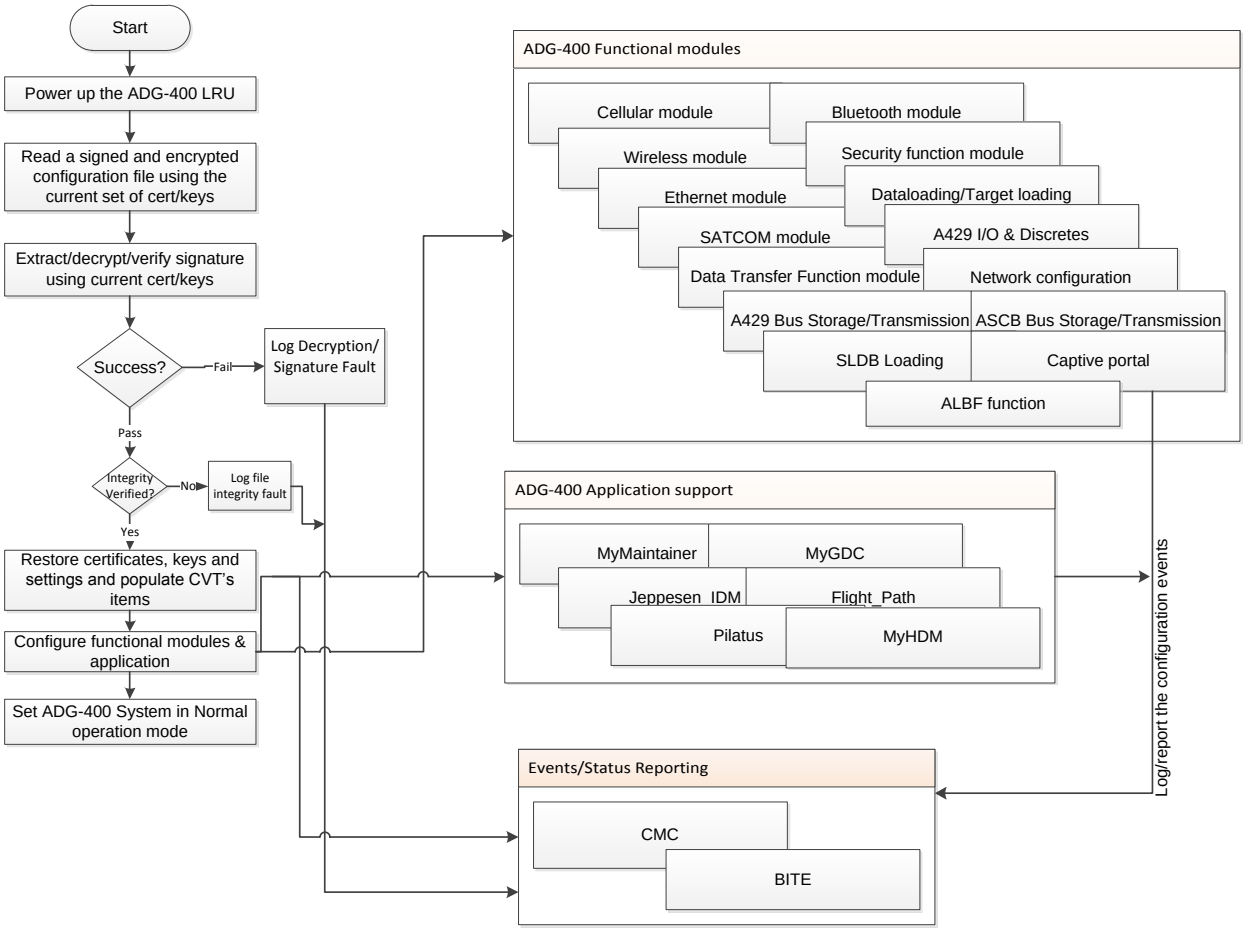
1AWObjectType : Description

The ADG-400 system read the NVM memory after power up. Upon reading, the ADG-400 system configures the Core module as well as any extended modules or S/W function modules or setup the applications support based on the fetched configuration data items. It also configures all of the required certificates/keys such that secure WiFi/Cellular/SATCOM connections or any other secure Ethernet connections can be established.

ID : SSS_2440

1AWObjectType : Description

Refer below diagram for the ADG-400 configuration operation after power up.



ID : SSS_147

1AWObjectType : Requirement

The ADG-400 **shall** read and extract the HCF, UCF and ACF configuration files following self-test.

ID : SSS_2408

1AWObjectType : Requirement

The ADG-400 **shall** verify the HCF, UCF and ACF configuration file signature, integrity and decrypt (HCF and UCF) on every power cycle.

Commentary: The purpose of this requirement is protecting encrypted files from discovery. Refer *Decrypt and Verify Signature Using Current Certificates and Keys* section for the details.

ID : SSS_150

1AWObjectType : Requirement

The ADG-400 **shall** retain the encrypted configuration file and restore certificates, keys and settings when power is cycled.

ID : SSS_151

1AWObjectType : Title

3.3.6.1.6 ADG-400 Factory Default Configuration

ID : SSS_152

1AWObjectType : Requirement

The Factory Default UCF of the ADG-400 **shall** contain no configuration.

ID : SSS_156

1AWObjectType : Requirement

The ADG-400 **shall** only use the Factory Default Certificates for HCF and UCF after a factory reset until the first operational HCF and UCF file is loaded.

Commentary: The ADG-400 receive their initial certificates during initial (factory) installation. Factory default certificates are embedded in the ADG-400 software.

ID : SSS_146

1AWObjectType : Title

3.3.6.1.7 ADG-400 Configuration After Factory Reset

ID : SSS_169

1AWObjectType : Requirement

The ADG-400 Factory Default State **shall** erase all user configuration data, logs, and operational data and leave the ADG-400 in Factory Default/ Baseline Configuration File state.

ID : SSS_170

1AWObjectType : Requirement

After performing a reset to factory defaults, the ADG-400 **shall** wait for the FACTORY RESET discrete to transition from ground to open before automatically rebooting the ADG-400.

ID : SSS_2449

1AWObjectType : Title

3.3.6.1.8 Providing Encapsulated Keys

ID : SSS_4826

1AWObjectType : Title

3.3.6.1.8.1 Decrypt and Verify Signature after a Factory Reset

ID : SSS_2458

1AWObjectType : Description

The Honeywell version of the ADG-400 Configuration Tool create a signed and encrypted HCF using a new set of certificates and keys which are encapsulated inside the Factory Default certificates and keys.

The customer version of the ADG-400 Configuration Tool will create a signed and encrypted UCF using a new set of certificates and keys which are encapsulated inside the Factory Default certificates and keys.

The Enterprise Portal create a signed and encrypted UCF using a new set of certificates and keys which are encapsulated inside the Factory Default certificates and keys. This method will be required if a factory reset has been performed. Note that this procedure can also be used to create a signed and encrypted UCF using new Current certificates and keys which are encapsulated inside the old Current certificates and keys.

The ACF contain parameters specific to an individual aircraft and upgradable features. The ACF file is created by aircraft manufactures and airlines via the Honeywell SIGNS Web Portal. The ACF is digitally signed for security purposes when received by aircraft manufactures and airlines from the Honeywell SIGNS Web Portal.

ID : SSS_2461

1AWObjectType : Requirement

The ADG-400 **shall** perform the following steps to read a signed and encrypted HCF using the factory default set of certificates and keys.

Steps:

1. Extract files from key_enc_sign_hcf_000000_factory.zip.

2. Decrypt the symmetric key [enc_symmetric.key] with the current Honeywell Private Encryption Key [FactoryDefaultHCFEncryptKey.private]. An Open SSL command-line example is: openssl smime -decrypt -binary -aes256 -in enc_symmetric.key -inform DER -out symmetric.key -inkey FactoryDefaultHCFEncryptKey.private -passin file:FactoryDefaultHCFEncryptKey.passphrase

3. Using the AES-256 algorithm, decrypt the enc_sign_hcf_000000_factory.zip file with the symmetric key [symmetric.key]. An Open SSL command-line example is: openssl enc -d -aes-256-cbc -md sha256 -in enc_sign_hcf_000000_factory.zip -out sign_hcf_000000_factory.zip -pass file:symmetric.key

4. Extract files from sign_hcf_000000_factory.zip.

5. Verify the extracted public certificate with Public Signing CA with CRL openssl verify -crl_check -CAfile HCFSigningCA.CER -CRLfile HCFSigningCA.CRL HCFGroundSigningKey.CER

6. Verify the extension of the Certificate HCFGroundSigningKey.CER:

    X509v3 Key Usage:

    Digital Signature, Non-Repudiation, Off-line CRL Signing

7. Verify the signature of the hash [nkey_hcf_000000_factory.bin] using the factory default Honeywell Public Signing Certificate [HCFGroundSigningKey.CER]. An Open SSL command-line example is:

openssl x509 -pubkey -noout -in HCFGroundSigningKey.CER > pubkey.pem openssl dgst -sha256 -verify pubkey.pem -signature nkey_hcf_000000_factory.bin nkey_hcf_000000_factory.zip

8. Extract nkey_hcf_000000_factory.zip and the extracted files will have current Honeywell Public Signing CA [HCFGroundSigningCA.CER], CRL [HCFGroundSigningCA.CRL], current Honeywell Private Encryption Key [HCFGroundEncryptKey.private], and current passphrase file [HCFGroundEncryptKey.passphrase] and all the HCF components.

Note:

The Current set of certificates and keys will be contained inside the factory default HCF.

ID : SSS_2462

1AWObjectType : Requirement

The ADG-400 **shall** perform the following steps to read a signed and encrypted UCF using the factory default set of certificates and keys.

Steps:

1. Extract files from key_enc_sign_ucf_000000_factory.zip.

2. Decrypt the symmetric key [enc_symmetric.key] with the current Customer Private Encryption Key [FactoryDefaultUCFEncryptKey.private]. An Open SSL command-line example is: openssl smime -decrypt -binary -aes256 -in enc_symmetric.key -inform DER -out symmetric.key -inkey FactoryDefaultUCFEncryptKey.private -passin file:FactoryDefaultUCFEncryptKey.passphrase

3. Using the AES-256 algorithm, decrypt the enc_sign_ucf_000000_factory.zip file with the symmetric key [symmetric.key]. An Open SSL command-line example is: openssl enc -d -aes-256-cbc -md sha256 -in enc_sign_ucf_000000_factory.zip -out sign_ucf_000000_factory.zip -pass file:symmetric.key

4. Extract files from sign_ucf_000000_factory.zip.

5. Verify the extracted public certificate with Public Signing CA with CRL openssl verify -crl_check -CAfile UCFSigningCA.CER -CRLfile UCFSigningCA.CRL UCFGroundSigningKey.CER

6. Verify the extension of the Certificate UCFGroundSigningKey.CER:

   X509v3 Key Usage:

   Digital Signature, Non-Repudiation, CRL Sign

7. Verify the signature of the hash [nkey_enc_sign_ucf_000000_factory.bin] using the factory default Customer Public Signing Certificate [UCFGroundSigningKey.CER]. An Open SSL command-line example is: openssl x509 -pubkey -noout -in UCFGroundSigningKey.CER > pubkey.pem openssl dgst -sha256 -verify pubkey.pem -signature nkey_ucf_000000_factory.bin nkey_ucf_000000_factory.zip

8. Extract nkey_enc_sign_ucf_000000_factory.zip. The extracted files will have current Customer Public Signing CA [UCFSigningCA.CER], CRL [UCFSigningCA.CRL], current Customer Private Encryption Key [UCFGroundEncryptKey.private] and current Customer Private Encryption Key [UCFGroundEncryptKey.passphrase] and UCF components.

Note:

The Current set of certificates and keys will be contained inside the factory default UCF.

ID : SSS_2460

1AWObjectType : Requirement

During ADG-400 HCF configuration installation, when all the Checks as mentioned in SSS_2355 are successful and the ADG-400 is in the configuration mode then the ADG-400 **shall** store the current set of Honeywell certificates and keys to non-removable media.

HCF Certificates:

- Honeywell Public Signing CA [HCFGroundSigningCA.CER],
- CRL [HCFGroundSigningCA.CRL],
- Current Honeywell Private Encryption Key [HCFGroundEncryptKey.private], and
- Current passphrase file [HCFGroundEncryptKey.passphrase]

ID : SSS_2463

1AWObjectType : Requirement

During ADG-400 UCF configuration installation, when all the Checks as mentioned in SSS_2355 are successful and the ADG-400 is in the configuration mode then the ADG-400 **shall** store the current set of Customer certificates and keys to non-removable media.

Customer Certificates:

- Current Customer Public Signing CA [UCFSigningCA.CER],
- CRL [UCFSigningCA.CRL],
- Current Customer Private Encryption Key [UCFGroundEncryptKey.private] and
- Current Customer Private Encryption Key [UCFGroundEncryptKey.passphrase].

ID : SSS_4827

1AWObjectType : Title

3.3.6.1.8.2 Decrypt and Verify Signature with Encapsulated Certificates and Keys

ID : SSS_2450

1AWObjectType : Description

The ADG-400 Configuration Tool or Enterprise Portal will create a signed and encrypted HCF/UCF using old Current certificates/keys (CurN-1) which are encapsulated inside the old Current certificates/keys (CurN). This will cause the ADG-400 to delete the oldest set of certificates/keys (CurN-1).

ID : SSS_2464

1AWObjectType : Requirement

During ADG-400 HCF configuration installation, the ADG-400 **shall** perform the following steps to decrypt and verify an HCF using old current set of certificates/keys (CurN-1) to obtain a new set of certificates/keys (CurN).

Steps:

1. Extract files from key_enc_sign_hcf_N.zip.

2. Decrypt the symmetric key [enc_symmetric.key] with the current Honeywell Private Encryption Key [GroundEncryptKey.private]. An Open SSL command-line example is: openssl smime -decrypt -binary -aes256 -in enc_symmetric.key -inform DER -out symmetric.key -inkey GroundEncryptKey.private --passin file:GroundEncryptKey.passphrase

3. Using the AES-256 algorithm, decrypt the enc_sign_hcf_N.zip file with the symmetric key [symmetric.key]. An Open SSL command-line example is: openssl enc -d -aes-256-cbc -md sha256 -in enc_sign_hcf_N.zip -out sign_hcf_N.zip -pass file:symmetric.key

4. Extract files from sign_hcf_N.zip.

5. Verify the extracted public certificate with Public Signing CA with CRL openssl verify -crl_check -CAfile HCFSigningCA.CER -CRLfile HCFSigningCA.CRL HCFGroundSigningKey.CER

6. Verify the extension of the Certificate HCFGroundSigningKey.CER:

   X509v3 Key Usage:

   Digital Signature, Non Repudiation

7. Verify the signature of the hash [nkey_hcf_N.bin] using the current Honeywell Public Signing Certificate [HCFGroundSigningKey.CER]. An Open SSL command-line example is: openssl x509 -pubkey -noout -in HCFGroundSigningKey.CER > pubkey.pem openssl dgst -sha256 -verify pubkey.pem -signature nkey_hcf_N.bin nkey_hcf_N.zip

8. Extract nkey_hcf_N.zip. The extracted files new Honeywell Public Signing CA [HCFGroundSigningCA.CER], CRL [HCFGroundSigningCA.CRL], new Honeywell Private Encryption Key [HCFGroundEncryptKey.private], and new passphrase file [HCFGroundEncryptKey.passphrase] and HCF components.

9. Verify the new GroundSigningKey.CER using SigningCA.CER and SigningCA.CRL. An Open SSL command-line example is openssl verify -crl_check -CRLfile SigningCA.CRL -CAfile SigningCA.CER GroundSigningKey.CER

Note:

This will cause the ADG-400 to delete the oldest set of certificates/keys (CurN-1).

ID : SSS_2455

1AWObjectType : Requirement

During ADG-400 UCF configuration installation, The ADG-400 **shall** perform the following steps to decrypt and verify a UCF using old current set of certificates/keys (CurN-1) to obtain a new set of certificates/keys (CurN).

Steps:

1. Extract files from key_enc_sign_ucf_N.zip.

2. Decrypt the symmetric key [enc_symmetric.key] with the current Customer Private Encryption Key [UCFGroundEncryptKey.private]. An Open SSL command-line example is: openssl smime -decrypt -binary -aes256 -in enc_symmetric.key -inform DER -out symmetric.key -inkey UCFGroundEncryptKey.private -passin file:UCFGroundEncryptKey.passphrase

3. Using the AES-256 algorithm, decrypt the enc_sign_ucf_N.zip file with the symmetric key [symmetric.key]. An Open SSL command-line example is: openssl enc -d -aes-256-cbc -md sha256 -in enc_sign_ucf_N.zip -out sign_ucf_N.zip -pass file:symmetric.key

4. Extract files from sign_ucf_N.zip.

5. Verify the extracted public certificate with Public Signing CA with CRL openssl verify -crl_check -CAfile UCFSigningCA.CER -CRLfile UCFSigningCA.CRL UCFGroundSigningKey.CER

6. Verify the extension of the Certificate UCFGroundSigningKey.CER:

X509v3 Key Usage:

Digital Signature, Non-Repudiation

7. Verify the signature of the hash [nkey_ucf_N.bin] using the current Customer Public Signing Certificate [GroundSigningKey.CER]. An Open SSL command-line example is: openssl x509 -pubkey -noout -in UCFGroundSigningKey.CER > pubkey.pem openssl dgst -sha256 -verify pubkey.pem -signature nkey_ucf_N.bin nkey_ucf_N.zip

8. Extract nkey_enc_sign_ucf_N.zip. The extracted files will have new Customer Public Signing CA [UCFGroundSigningCA.CER], CRL [UCFGroundSigningCA.CRL], new Customer Private Encryption Key [UCFGroundEncryptKey.private], and new passphrase file [UCFGroundEncryptKey.passphrase] and UCF components.

Note:

The ADG-400 to delete the oldest set of certificates/keys (CurN-1).

ID : SSS_2457

1AWObjectType : Requirement

During ADG-400 HCF/UCF configuration installation in the configuration mode, after successful decryption and verification, if zip name starts with nkey then ADG400 **shall** replace the decryption and verification certificates of the ADG-400 box with the certificates present in the config folder of nkey_hcf/ucf_N.zip

ID : SSS_4836

1AWObjectType : Title

3.3.6.1.9 Validating & Storing the CRLs against the CA Certificate

ID : SSS_4837

1AWObjectType : Description

Certificate Revocation List (CRLs) must be frequently downloaded to keep the list updated at the ADG-400. ADG-400 device should fetch the CRLs from the ground server to the sync area designated for the CRLs stored on the device. Once synced, files should be moved from the sync area to the operational area based on the requirements stated in this section.

This section contain the requirements on CRLs validation and replacing the current ones with the new ones.

ID : SSS_4838

1AWObjectType : Requirement

When CRLs Data Transfer rule is configured, the ADG-400 **shall** fetch the CRLs from the ground server to the sync area designated for the CRLs stored on the device.

Sync area folders are

1. CRL/radiusserver/wifiClientCA.CRL
2. CRL/sslserver/standardClientSSLCA.CRL
3. CRL/config/HCFSigningCA.CRL
4. CRL/verification/SigningCA.CRL
5. CRL/config/UCFSigningCA.CRL

ID : SSS_4839

1AWObjectType : Requirement

The ADG-400 **shall** perform CRL validation against the CA Certificate if it finds a new CRL in the respective directory at power up.

ID : SSS_4842

1AWObjectType : Requirement

The ADG-400 **shall** obtain new CRL number from the CRL and compare it against the CRL number of the current CRL for determining if it is new CRL or not.

ID : SSS_4844

1AWObjectType : Requirement

Detected during CRL validation, by the ADG-400:

- Unsuccessful CRL Validation event

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF2 SECURITY

L3: 0x01 CRL VALIDATION FAILURE

L4.Type: 01b EVENT

L4.Action: 0b N/A

L4.Level: 001b WARNING

L4.Hist: 010b SECURITY

L4.Detect: 000b CONTINUOUS

Additional Text: one or more parameters, in the following form:

- <parameterlist> = <parameter> [';' <parameter>]

  <parameter> = CRL name

Detection Action: RECORD

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_4843

1AWObjectType : Requirement

CRL Validation Error **shall** be detected by the ADG-400 whenever CRL validation is unsuccessful due to the error.

ID : SSS_5208

1AWObjectType : Requirement

Detected during CRL validation, by the ADG-400:

- Successful CRL validation event

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF2 SECURITY

L3: 0x02 CRL VALIDATION SUCCESSFUL

L4.Type: 01b EVENT

L4.Action: 0b N/A

L4.Level: 00b INFORMATION

L4.Hist: 010b SECURITY

L4.Detect: 000b CONTINUOUS

Additional Text: one or more parameters, in the following form:

- <parameterlist> = <parameter> [';' <parameter>]

  <parameter> = CRL name

Detection Action: RECORD

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_4841

1AWObjectType : Requirement

If the verification is successful and the CRL number of the CRL in the CRL directory is greater than the CRL number of the current CRL (located in the ADG-400 secure non-volatile memory), the ADG-400 **shall** replace its current CRL (operational location) with the new CRL.

ID : SSS_2456

1AWObjectType : Title

3.3.6.1.10 Configuration Installation, Decryption & Signature F&E Logging

ID : SSS_2451

1AWObjectType : Requirement

Informational error detected during HCF/UCF/ACF configuration installation, by the ADG-400

- No such file or directory
- Configuration File Signature Failure
- Configuration File Decryption Failure
- File Integrity Failure
- Version Same

- Configuration Installation Success

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF7 CONFIGURATION MANAGEMENT

L3: 0x10 NO SUCH FILE OR DIRECTORY

0x11 CONFIGURATION FILE SIGNATURE FAILURE

0x13 CONFIGURATION FILE DECRYPTION FAILURE

0x15 FILE INTEGRITY FAILURE

0x17 VERSION SAME

0x18 CONFIGURATION INSTALLATION SUCCESS SUCCESS

L4.Type: 01b EVENT

L4.Action: 0b N/A

L4.Level: 00b INFORMATION

L4.Hist: 001b OPERATIONAL

L4.Detect: 000b CONTINUOUS

- Additional Text:
  - For L3=0x18, <parameter> = <filename> <fileversion>
  - For L3=11 and 13, <parameter> = <filename> <cert/keyname>
  - For others L3, <parameter> = <filename>;<calculated digest>;<stored digest>

  where
  - <filename> = name of the configuration file,
  - <cert/keyname> is the name of cert/key used for decryption and signature verification &
  - <fileversion> is the version of the installed configuration file

Detection Action: RECORD

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_4830

1AWObjectType : Requirement

Critical error detected during HCF/UCF/ACF Decryption/Signature verification at initialization after power up, by the ADG-400

- Configuration File Signature Failure
- Configuration File Decryption Failure
- File Integrity Failure

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF7 CONFIGURATION MANAGEMENT

L3: 0x12 CONFIGURATION FILE SIGNATURE FAILURE

0x14 CONFIGURATION FILE DECRYPTION FAILURE

0x16 FILE INTEGRITY FAILURE

L4.Type: 00b FAULT

L4.Action: 1b SET

L4.Level: 11b CRITICAL

L4.Hist: 010b SECURITY

L4.Detect: 001b POST

Additional text:

- For L3=0x16, <parameter> = <filename>;<calculated digest>;<stored digest>
- For others, <parameter> = <filename> <cert/keyname>

Where <filename> = name of the configuration file, <cert/keyname> is the name of cert/key used for decryption and signature verification

Detection Action: RECORD;ASSERT 10,24;DEASSERT 8;REPORT TO CMC ASSERT 0;CRITICAL MODE

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_2452

1AWObjectType : Requirement

The ADG-400 **shall** detect the "No such file or directory" error when configuration file installation is initiated using "site install filepath/filename" and the file or directory is not found.

ID : SSS_2447

1AWObjectType : Requirement

During ADG-400 HCF/UCF configuration installation or during initialization after power up, the ADG-400 **shall** detect the HCF/UCF Configuration File Decryption Failure with the configuration file name when it fails to decrypt the encrypted & signed HCF/UCF file with the symmetric key using AES-256 algorithm.

ID : SSS_2448

1AWObjectType : Requirement

During ADG-400 HCF/ACF/UCF configuration installation or during initialization after power up, the ADG-400 **shall** detect the HCF/ACF/UCF Configuration File Signature Failure with the configuration file name when it fails to verify the signature of the hash using the Public Signing Certificate.

ID : SSS_4223

1AWObjectType : Requirement

During ADG-400 HCF/ACF/UCF configuration installation, the ADG-400 **shall** detect the file Version Same error with the configuration file name when the digest in the candidate header file is equal to digest in currently installed header file.

ID : SSS_4224

1AWObjectType : Requirement

The ADG-400 **shall** detect the installation success events when candidate file is validated and successfully installed.

ID : SSS_5084

1AWObjectType : Requirement

Informational error detected during ADG-400 HCF/ACF/UCF configuration installation, by the ADG-400

- HCF Configuration File Missing
- ACF Configuration File Missing
- UCF Configuration File Missing

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF7 CONFIGURATION MANAGEMENT

L3: 0x19 HCF CONFIGURATION FILE MISSING

0x1A ACF CONFIGURATION FILE MISSING

0x1B UCF CONFIGURATION FILE MISSING

L4.Type: 01b EVENT

L4.Level: 00b INFORMATION

L4.Action: 0b N/A

L4.Hist: 001b OPERATIONAL

L4.Detect: 000b CONTINUOUS

Additional text: Configuration file list missing

Detection Action: RECORD

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_5083

1AWObjectType : Requirement

Critical error detected at initialization after power up, by the ADG-400

- HCF Configuration File Missing
- ACF Configuration File Missing
- UCF Configuration File Missing

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF7 CONFIGURATION MANAGEMENT

L3: 0x1C HCF CONFIGURATION FILE MISSING

0x1D ACF CONFIGURATION FILE MISSING

0x1E UCF CONFIGURATION FILE MISSING

L4.Type: 00b FAULT

L4.Action: 1b SET

L4.Level: 11b CRITICAL

L4.Hist: 010b SECURITY

L4.Detect: 001b POST

Additional text: Configuration file list missing

Detection Action: RECORD;ASSERT 10,24;DEASSERT 8;REPORT TO CMC ASSERT 0;CRITICAL MODE

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_4230

1AWObjectType : Requirement

The ADG-400 **shall** detect the file integrity fault when digest stored in the candidate header file and the calculated digest does not match during:

- ADG-400 HCF/ACF/UCF configuration installation, OR
- ADG-400 initialization after power up

ID : SSS_214

1AWObjectType : Title

3.3.6.1.11 ADG-400 Configurable Functions

ID : SSS_660

1AWObjectType : Title

3.3.6.1.11.1 Hardware Configuration

ID : SSS_645

1AWObjectType : Requirement

ADG-400 **shall** read the hardware part number from non volatile memory and control the hardware configurations available for various platforms as per below table:

Configuration Module	Hardware Part Number					
	Config 1	Config 1A	Config 1B	Config 2A	Config 2B	Config 4
ADG-400 Core Unit	✓	✓	✓	✓	✓	✓
Wi-Fi and Bluetooth Module	✓	✓	✓	✓	✓	✓
Wi-Fi External Module	✓	✓	✓	✓	✓	✓
Cellular Module	✓	✓	✓	✓	✓	✓
RS Module		✓				
Ethernet Module			✓			
ASCB Module					✓	
ARINC Module						✓
RSU				✓	✓	✓

ID : SSS_2333

1AWObjectType : Description

After completion of POST, The ADG-400 monitor, report and log the critical fault when H/W part number included in the HCF configuration is different than the actual H/W part number available from platform.

The ADG-400 should read the actual H/W part number data from the SSD memory and compare with the HCF field. In case, HCF field is blank or not included then also ADG-400 should raise the critical fault for missing configuration date.

ID : SSS_653

1AWObjectType : Title

3.3.6.1.11.2 Network Configuration

ID : SSS_4

1AWObjectType : Description

This section describes the network configuration details of the ADG-400.

ID : SSS_2437

1AWObjectType : Commentary

The UCF configuration allows the user to control the LAN test. It also contains the various hostnames and the IP addresses which in addition to those determined by the MLF, confg_ca.csv or already defined in the requirement.

ID : SSS_2436

1AWObjectType : Requirement

The ADG-400 **shall** support the LAN test of the LAN connection only if enabled in the UCF configuration file.

ID : SSS_658

1AWObjectType : Requirement

The ADG-400 **shall** test the LAN connection to the NICs, AGMs, CMC and Printer when requested through the FTP command site lantest filename, and create a comma separated value file with the results of the LAN test.

- NIC, DUNIC, RNIC, AGM addresses to be determined from MLF.
- CMC address from config_caf.csv maintenance row.
- Printer address use 192.168.200.5
- In addition, read host addresses and hostname as configured in the UCF configuration file.
    - Determine the router's LAN IP address of default gateway for the all the IP addresses.
    - Ping the IP address of the default gateway to verify that the default gateway is functioning and ADG-400 can communicate with a local host on the local network.
        - if pings successful,
            - Create a thread for each address to allow parallel testing.
            - Ping each address as follows:
                - 50 packets
                - 1400 byte per packet
                - Time out 1 second
                - Delay 500 msec between packets
            - Record results for each address
        - if pings fail, then record results for each default gateway in the below format:

        <Default gateway IP address> : <failure message>

Example file content:

Address,Hostname,%Packetslost

192.168.1.1,nic1,0

192.168.1.33,nic2,0

192.168.1.2,rnic1,0

192.168.1,34,rnic2,0

192.168.200.1,cmc,10

192.168.200,5,printer,50

10.10.10.20: Destination host unreachable

ID : SSS_659

1AWObjectType : Commentary

The ADG-400 will return 200 OK in response to the site lantest command immediately, however the test could take up to a minute to complete before the file is available

ID : SSS_2435

1AWObjectType : Requirement

The ADG-400 **shall** support the LAN test of the LAN connection only if enabled in the UCF configuration file.

ID : SSS_2494

1AWObjectType : Requirement

When detected during the testing of the LAN connection to the NICs , CMC and Printer when requested through the FTP command site lantest, by the ADG-400:

- Packet loss is exceeds 5% during the LAN test

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0xF4 EPIC MEMBER SYSTEM MGMT

L3: 0x06 LAN CONNECTIVITY FAULT

L4.Type: 00b EVENT

L4.Action: 0b N/A

L4.Level: 01b WARNING

L4.Hist: 001b OPERATIONAL

L4.Detect: 000b CONTINUOUS

Detection Action: RECORD;REPORT TO CMC ASSERT 4

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_162

1AWObjectType : Title

3.3.6.1.11.3 Non Volatile Memory (SSD) Partitioned & Size Configuration

ID : SSS_2409

1AWObjectType : Description

This section describes the SSD partitioned configuration details of the ADG-400 for the various services.

ID : SSS_2410

1AWObjectType : Requirement

The ADG-400 OS **shall** configure the partition size for all the required partitions (except primary, secondary and F&E partitions) based on the partition and size information available in the HCF configuration.

ID : SSS_2411

1AWObjectType : Requirement

The ADG-400 OS **shall** configure the partition size with the default size for the partition which are not configured in the HCF configuration as specified under **ADG-400 Storage Memory Allocations**.

ID : SSS_299

1AWObjectType : Title

3.3.6.1.11.4 Radio Configuration

ID : SSS_45

1AWObjectType : Description

This section describes the radio configuration details of the ADG-400.

ID : SSS_297

1AWObjectType : Title

3.3.6.1.11.5 Wi-Fi Power Configuration

ID : SSS_52

1AWObjectType : Description

Refer **Network Manager- RF Switching** section for the Wi-Fi Power management details.

ID : SSS_57

1AWObjectType : Title

3.3.6.1.11.6 Security/Network Firewalling Configuration

ID : SSS_51

1AWObjectType : Description

This section describes the security/network firewalling configuration details of the ADG-400.

ID : SSS_2334

1AWObjectType : Requirement

The ADG-400 **shall** provide authorized access to its storage area to the mobile device application based on the area & authorization key defined in configuration file when secured connected mobile device application want to access the identified area with valid username: password.

ID : SSS_2335

1AWObjectType : Requirement

The ADG-400 **shall** use the shorewall.zip file under the HCF configuration file and configure the gateway/firewall and the Linux network subsystem.

ID : SSS_644

1AWObjectType : Requirement

The ADG-400 HCF, UCF, and ACF **shall** support PKI used for signature verification.

ID : SSS_646

1AWObjectType : Requirement

The ADG-400 firewall rules **shall** be configurable only by Honeywell as a part of the HCF.

ID : SSS_2395

1AWObjectType : Requirement

The ADG-400 firewall rules rules **shall** be considered a Parameter Data Item(PDI).

ID : SSS_58

1AWObjectType : Title

3.3.6.1.11.7 SIM Card Configuration

ID : SSS_53

1AWObjectType : Description

This section describes the SIM card configuration details of the ADG-400.

ID : SSS_1242

1AWObjectType : Requirement

The ADG-400 **shall** attempt the initial cellular connection for gathering time and country information when atleast one of the following options is configured in the ADG-400 configuration file:

- default flag is set to true, OR
- country code is set to "AAA"

ID : SSS_1253

1AWObjectType : Requirement

After determination of the aircraft location, The ADG-400 **shall** switch to an alternate cellular radio as determined by the ADG-400 configuration file.

ID : SSS_7

1AWObjectType : Title

3.3.6.1.11.8 Data Transfer Function Configuration

ID : SSS_54

1AWObjectType : Description

Refer Data Transfer section for the data transfer configuration details of the ADG-400.

ID : SSS_298

1AWObjectType : Title

3.3.6.1.11.9 Data Loading Configuration

ID : SSS_55

1AWObjectType : Description

Refer Data Loading section for the data loading configuration details of the ADG-400.

ID : SSS_1185

1AWObjectType : Title

3.3.6.1.11.10 Data Bus Recording Configuration

ID : SSS_1186

1AWObjectType : Description

Refer Data Bus Recording section that describes the data bus recording configuration details of the ADG-400.

ID : SSS_48

1AWObjectType : Title

3.3.6.1.11.11 CMC Configuration

ID : SSS_9

1AWObjectType : Description

Refer CMC Interface section for the CMC configuration details of the ADG-400.

ID : SSS_47

1AWObjectType : Title

3.3.6.1.11.12 ADG-400 Access Point Mode SSID Configuration on the Avionics side.

ID : SSS_56

1AWObjectType : Description

This section describes the WAP SSID configuration details of the ADG-400.

The SSID is a unique identifier that wireless network devices use to establish and maintain wireless connectivity. This is done in order to prevent unauthorized users to access a ADG-400 wireless network through an access point that has a default SSID and no security settings. ADG-400 must create an SSID before anyone can enable the access point radio interfaces.

Refer SSS_4208 for the format of the SSID.

ID : SSS_4462

1AWObjectType : Description

The configuration files will contain a X.509 compliant digital certificate for authentication and authorization of the WPA2 security protocols for both connecting to ground based 802.11 networks as well as allowing 802.11 client devices to connect to the ADG-400 802.11 a/b/g/n/ac access point (AP).

The wireless LAN provides wireless connectivity to maintenance applications. Enforcement of which wireless users can use the wireless LAN system is performed by the ADG-400 which hosts an On-board Authentication Server that supports the RADIUS protocol with WPA2 extensions.

ID : SSS_1192

1AWObjectType : Requirement

When configured for Epic platform, after completion of POST, when tail number, as determined from EPIC Member System Function, matches with the one define in the ACF configuration file, the ADG-400 **shall** set the wireless access point SSID in a format defined in the UCF configuration file.

NOTE:

- If there is no format define in the UCF file then use the tail number for setting up the SSID.
- Tail number is determined from the EPIC Member System Function.

ID : SSS_1193

1AWObjectType : Requirement

When detected, during POST, by the ADG-400:

- ADG-400 configured for an EPIC platform. AND
- Tail number mismatch between the value determined from the EPIC Member System Function and the tail number defined under the aircraft setting in ACF and UCF.

The ADG-400 **shall** handle as follows:

L1: 0x01 ADG-400

L2: 0x02 ADG-400 CONTROLLER

L3: 0x01 TAIL NUMBER MISMATCH

L4.Type: 00b FAULT

L4.Action: 1b SET

L4.Level: 11b CRITICAL

L4.Hist: 011b FAULT

L4.Detect: 001b POST

Detection Action: RECORD;ASSERT 10,24;DEASSERT 8;REPORT TO CMC ASSERT 0,12;CRITICAL MODE

Confirmation: N/A

Confirmation Action: N/A

Recovery: N/A

Recovery Action: N/A

ID : SSS_705

1AWObjectType : Requirement

The ADG-400 **shall** suppress broadcasting of its SSID when the broadcast SSID parameter is set to false under ADG-400 access point mode settings in the UCF configuration file.

ID : SSS_4463

1AWObjectType : Requirement

The ADG-400 **shall** perform the configuration settings for enabling the RADIUS accounting, authentication and authorization for this SSID.

ID : SSS_4464

1AWObjectType : Requirement

The ADG-400 **shall** use X.509 compliant digital certificate from the HCF configuration file for authentication and authorization of the WPA2 security protocols for both connecting to ground based 802.11 networks as well as allowing 802.11 client devices to connect to the ADG-400 802.11 a/b/g/n/ac access point (AP).

ID : SSS_4465

1AWObjectType : Description

Refer **Wireless LAN Security** section under Cyber Security for the requirements on the ADG-400 RADIUS accounting, authentication and authorization using X.509 compliant digital certificate.

ID : SSS_49

1AWObjectType : Title

3.3.6.1.11.13 Bluetooth Configuration

ID : SSS_8

1AWObjectType : Description

This section describes the Bluetooth configuration details of the ADG-400.

ID : SSS_282

1AWObjectType : Title

3.3.6.1.11.14 Automatic Activity Log File Generation

ID : SSS_358

1AWObjectType : Description

The ADG-400 will create a copy of the activity log file for automated transfer. The automatic activity log file creation will be controlled by a setting in the UCF file. On a transition from air to ground the ADG-400 will overwrite the existing copy of the file (slot1/status/actStatus.log). When auto_getlog_enable parameter under fleetsetting file is 'true' then automatic activity log file generation is consdered enable.

ID : SSS_357

1AWObjectType : Requirement

When the ADG-400 UCF confguration data indicates that the automatic activity log file generation is enable, the ADG-400 **shall** copy the activity log to SD card slot1/status/actStatus.log when the operational mode transitions from air to ground.

ID : SSS_355

1AWObjectType : Title

3.3.6.1.11.15 Wi-Fi In-Air Operation

ID : SSS_354

1AWObjectType : Description

The ADG-400 will use an operator selectable option in the configuration file and ALBF data to allow operation of the Wi-Fi in air. The ALBF data will indicate safe modes of operation, such as ground and cruise for enabling the Wi-Fi. If the option is not selected or ALBF is not available the ADG-400 will use the On Ground logic.

Note: The ALBF data safe modes of operation will only be used for Wi-Fi operation, all other ADG-400 operation using air/ground logic will be as per SSS_5196.

ID : SSS_359

1AWObjectType : Description

Refer **Network Manager- RF Switching** section for the Wi-Fi in air operation details.

ID : SSS_706

1AWObjectType : Title

3.3.6.1.11.16 Remote Terminal/Apps Connection Configuration

ID : SSS_714

1AWObjectType : Requirement

The ADG-400 **shall** limit the number of erroneous connections to a connect fault value specified in the ADG-400 ACF configuration file.

ID : SSS_715

1AWObjectType : Requirement

The ADG-400 **shall** limit the number of erroneous authentication to a authentication fault value specified in the ADG-400 ACF configuration file.

ID : SSS_716

1AWObjectType : Requirement

The ADG-400 **shall** employ a timeout value as specified in a External App timeout parameter in the ADG-400 ACF configuration file and declare heartbeat message failure if the time between External App heartbeat messages reception exceeds the External App timeout value.

ID : SSS_245

1AWObjectType : Title

3.3.6.2 Data Transfer

ID : SSS_32

1AWObjectType : Description

The Data Transfer Function defines the automatic transfer of data files on and off the aircraft in a secured environment. ADG-400 uses the Wi-Fi or cellular or SATCOM connectivity to ground based access points to enable automatic data transfers through the internet to and from the data center servers. The configuration files defines radio selection based on aircraft state, network, ground server, addresses, storage directory and authentication details for logon, connectivity and data transfer. The transferred files will be signed and encrypted for security considerations. Refer to **Figure - Data Transfer** for System Level depiction of Data Transfer Function.

ID : SSS_1190

1AWObjectType : Title

3.3.6.2.1 Figure - Data Transfer

ID : SSS_1194

1AWObjectType : Figure