cid:image001.jpg@01D4201D.1D653760

TINETZ / IKB

Honeywell / Security Suite 1 Adoption Impact

Author(s): Tom McDonna

Team: OMNETRIC Solution Architecture

Version : 0.2

Date: August 2018

Status: Draft

Accessibility: **Restricted**

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Revisions** |
| 10/08/18 | 0.1 | T McDonna | Initial copy |
| 16/08/18 | 0.2 | T McDonna | Frame Counter added |
| 20/08/18 | 0.3 | T McDonna | Changes and revisions following first UDIS call |

# References

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Document Title** | **Author** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# Table of Contents

[Revisions 2](#_Toc522618686)

[References 2](#_Toc522618687)

[Table of Contents 3](#_Toc522618688)

[1 Introduction 4](#_Toc522618689)

[1.1 Overview 4](#_Toc522618690)

[1.2 Purpose 4](#_Toc522618691)

[1.3 Scope 4](#_Toc522618692)

[2 Development Areas 5](#_Toc522618693)

[2.1 HLS\_SECRET (HLS6) Support 6](#_Toc522618694)

[2.1.1 Key Import 6](#_Toc522618695)

[2.1.2 Key Export 6](#_Toc522618696)

[2.1.3 Lifecycle Management / Key Renewal 7](#_Toc522618697)

[2.2 PSK Support 8](#_Toc522618698)

[2.2.1 Key Import 8](#_Toc522618699)

[2.2.2 PSK Lifecycle Management 8](#_Toc522618700)

[2.2.3 PSK Replacement Workflow 8](#_Toc522618701)

[2.3 Security Suite 1 - Asymmetric Key Support 9](#_Toc522618702)

[2.3.1 Meter CSR (at Manufacture) 9](#_Toc522618703)

[2.3.2 Meter Certificate Import 9](#_Toc522618704)

[2.3.3 Meter Asymmetric COSEM Key Lifecycle Management 10](#_Toc522618705)

[2.3.4 Meter to HES Client Certificate Mapping 11](#_Toc522618706)

[2.3.5 HES Client Certificate Lifecycle Management 12](#_Toc522618707)

[2.4 Security Suite 1 - HLS7 Authentication 13](#_Toc522618708)

[2.5 Key Agreement Key Exchange Mechanism 14](#_Toc522618709)

[2.6 Frame Counter Handling 16](#_Toc522618710)

# Introduction

## Overview

## Purpose

## Scope

# Development Areas

The following section details the potential development work required to enable the support of Honeywell / Security Suite 1 devices. The descriptions give background information and specific requirements for development.

Key:

Where a line contains a keyword in red / bold it is an expected development item. Ex: **Extend** | **New**

Where a line contains a keyword in orange / bold it is a potential development, subject to further confirmation. Ex: **Extend** | **New**

## HLS\_SECRET (HLS6) Support

### Key Import

***Impact: MDM | KMS***

Honeywell meters require HLS6 authentication for Local connections. This means that they have an additional Key installed in Memory and provided to the Siemens Solution via Shipment File.

We expect to receive an HLS\_SECRET key for three Clients; Installation, Certification and Maintenance.

The HLS\_SECRET Keys will be delivered in the same Shipment File as other Keys. It will use the same Transport Key as other Symmetric Keys.

The Worldline JSS is going to be modified to include a new ‘ProtectedSessionKeyCapability’ for the HLS\_SECRET. This is in part because the definition of the HLS\_SECRET is such that it may require padding, the purpose of the new KeyCapability is to standardize that padding (TBD with Worldline). The JSS function will strip off any PKCS#5 padding added to the HLS\_SECRET and replace with the correct padding. The nature of the padding is currently under discussion between Worldline and Honeywell, it has no impact to KMS.

This Key type will require its own Key Templates and should fall under the existing Key Issuer / Template model, however an extension may be required:

**Development areas are likely to be:**

* KMS / MDM - We must **Extend** the existing Shipment File and Key Import processes to support Import of this Key Type.
* KMS - Because renewal of the HLS\_SECRET Keys requires that the new Secret is wrapped with the Meter Master Key (ie Key Wrap is not Role based) we may need to **Extend** existing Key Template configuration to add a new Role of ‘Meter Master’ or something similar.

### Key Export

***Impact: KMS***

The HLS\_SECRET described previously needs to be exportable for provision to Field Tools / Hand Held devices.

The key should be wrapped in the same way as other symmetric meter keys, as described in the Import section, the JSS will ensure that the length of the Key is always padded to enable a Key Wrapped export.

**Development areas are likely to be:**

* KMS - We need to **Extend** the existing getKey interface to allow specification of the keyType HLS\_SECRET to trigger the export of the wrapped HLS\_SECRET key for the specific Role.

### Lifecycle Management / Key Renewal

***Impact: KMS | MDM | HES***

The HLS\_SECRET should be subject to the same lifecycle management as other symmetric meter keys. Replacement of the HLS\_SECRET should be included in:

* Scheduled (time / expiry based) lifecycle management
* Implicit / Explicit Key Renewal REST Services requests

The HLS Secret must be replaced using Key Wrapping mechanisms, the wrap key that must be used is the Meter Master Key.

**Development areas are likely to be:**

* MDM - **Extend** Lifecycle Management (automatic Job) to include monitoring of HLS\_SECRET
* MDM – **Extend** the Key Renewal Workflow to cater for the process flow required for HLS\_SECRET
* KMS - **Extend** Explicit Key Renewal REST Definition (explicitKeyRenewal) to include specification of the HLS\_SECRET keyType
* KMS - **Extend** the Implicit Key Renewal REST request to include the HLS\_SECRET keyType
* KMS - **Extend** the Key Template definition to allow specification that replacement HLS\_SECRET keys should be wrapped with the Meter KEK.
* KMS – **Extend** getDlmsKeyForRenewal to include keyType HLS\_SECRET. The new Key generation for HLS\_SECRET must ensure the following is true:
  + Worldline JSS Documentation - “use cosemSecretRenewal(ProtectedSessionKeyCapability, int, ProtectedSessionKey, KeyLabel) with ProtectedSessionKeyCapability.SM\_WK\_CRYPTAUTH\_RENEWAL and a secret length of 16 or 32”
* UDIS – UDIS will then need to receive the wrapped HLS\_SECRET from KMS in a UAA MeterUpdate, construct an appropriate APDU this may be a **New** APDU definition
* KMS – **Extend** keyUpdate service definition to include specification of keyType HLS\_SECRET, to allow UDIS to notify KMS of the successful replacement.

The renewal process related to several of the items in the list above is based around following the outline process here: <https://devops-omnetric.ptdea.siemens.com:446/display/SDRE1/Key+Renewal>

## PSK Support

### Key Import

***Impact: KMS* | *MDM* | *HES***

During manufacture a PSK and PSK\_KEK will be installed in the Meter. These will be provided to KMS via Shipment File alongside all other Symmetric Keys, we need to import them to KMS.

### PSK Lifecycle Management

KMS must monitor PSK (and maybe PSK KEK) lifecycle, triggering a renewal process appropriately.

### PSK Replacement Workflow

The replacement workflow for PSK notionally follows the generic PSK replacement workflow, with one notable exception.

In some circumstances, the new PSK must also be pushed to Connexo, wrapped with the Connexo DB Storage Key. These circumstances are:

* When successful PSK exchange is explicitly confirmed to MDM
* When the success status of the PSK exchange is unknown (ie no response / timeout)

## Security Suite 1 - Asymmetric Key Support

### Meter CSR (at Manufacture)

***(Potential) Impact: KMS* | *Connexo***

In order for Honeywell to obtain the COSEM Device Certificates required to manufacture the Meters (one per Meter) the Meter must generate a Certificate Signing Request that should be submitted to the Meter ECDSA Sub-CA in the PKI of the Siemens Solution.

The manufacturing process results in a ZIP file containing batches of Certificate Signing Requests. A process must consume these files and submit the CSRs to the PKI, capturing the resultant certificate so it can be provided back into the manufacturing process for installation in the Meter. There are three options for processing of this file:

1. Connexo – Consumes the Zip, submits the CSR and gathers the certificates. It is believed that there is already a near-identical process in place to support manufacture of Gateway devices.
2. KMS – Create a **New** feature in KMS to allow consumption of the zip file, submit the CSRs and gather the resultant certificates. The format of the Zip file and resultant Certificate File would need to be established. Siemens would also insist on application of security controls to the Zip file (a digitial signature, following the same structure as the shipment file signature.
3. Manual / Direct 1 – Use standard EJBCA functionality to process the batch, reliant on EJBCA providing appropriate interfaces.
4. Manual / Direct 2 –Develop a **New** application outside Connexo or KMS to consume the zip file and submit the CSRs to the PKI via the Web Service interfaces and package the resultant certificates appropriately.

Option #1 is the preference of both Siemens and TIN / IKB. Development responsibility for Option #4 (if required) could be with Siemens or Honeywell.

### Meter Certificate Import

***Impact: KMS | MDM***

During the Meter manufacturing process a per Meter COSEM Certificate is created and installed in the Meter. KMS requires knowledge of this Certificate in order to perform several cryptographic operations to support Security Suite 1 and HLS7. The expectation is that these Certificates will be provided to the Siemens Solution via Shipment File.

This Shipment File will be signed in the same way as Shipment Files for Symmetric meter keys.

The format of the Shipment File is TBD, two options proposed are:

1. Using the Gateway Shipment File Format
2. Extending the DSMR 4.0 XSD to add Certificate capability

**Development areas are likely to be:**

* MDM - **Extend** the Shipment File Loader process to process the Certificate File for Meters, including making per certificate requests to the shipmentFileUploadCertificates service.
* KMS - **Extend** the shipmentFileUploadCertificates to include deviceType = Meter (this may already be in place, hence the orange marking, see <https://devops-omnetric.ptdea.siemens.com:446/display/ATM/Certificate+Import+-+REST+Service#space-menu-link-content>)

### Meter Asymmetric COSEM Key Lifecycle Management

***Impact: KMS | MDM | HES***

KMS Lifecycle Management will be responsible for monitoring impending replacement schedules for the Asymmetric meter keys. The KMS ability to do this is based around Lifecycle Management monitoring the lifetime of the Meter Certificates it must have imported to support asymmetric cryptograohic operations. The mechanism for identification of Keys requiring renewal should be similar to the existing process for symmetric meter keys and asymmetric gateway TLS keys, with lifecycle paramaters defined via Certificate Template.

The process will roughly follow:

1. KMS Lifecycle Management identifies an asymmetric meter key requires renewal, initiates MDM workflow.
2. MDM workflow initiates UDIS workflow via UAA interface
3. UDIS initiates Meter process, resulting in a CSR submitted to KMS
4. KMS submits CSR to PKI, stores response certificate, returns response certificate to UDIS
5. UDIS updates meter, and notifies KMS of success / failure status.

**Development areas are likely to be:**

* KMS - **Extend** Lifecycle Management to include monitoring of Meter Asymmetric Keys / Certificates
* MDM - Create a **New** MDM Gateway Certificate Renewal Workflow to handle the initiation to UDIS of Meter Certifcates renewal requests.
* MDM - **Extend** the parameter definition for the UAA UDIS interface to include a type that indicates to UDIS it must initiate an Asymmetric key renewal, the new MDM workflow will use this.
* UDIS - Create a **New** UDIS workflow to orchestrate the UDIS -> Meter interactions
* KMS - Create a **New** KMS Service Interface, or **Extend** the /cert/sign interface to permit submission of the Meter CSR to PKI via KMS. Meter CSRs must be submitted to a different PKI and the resultant Certificate handled differently for gateway and meter CSRs, so provision must be made for the service to determine the difference.

### Meter to HES Client Certificate Mapping

***Impact: KMS | MDM***

To facilitate Security Suite 1 functionality and enable the manufacturing process to commence, the Siemens Solution must provide a COSEM Certificate for installation in the Meter for each of the Roles below:

* Management Client
* Readout Client
* PLC Management Client

To clarify subsequently used terminology, these certificates are called ‘HES Client Certificates’.

The high level process surrounding the generation of the HES Client Certificates is something like:

1. Order an appropriate Key from Worldline, one for each Role, per customer (ie 6 total), each key will be given an alias.
2. Import Keys to HSM
3. Use Command Line JSS Tool to generate a Certificate Signing Request
4. Process the CSR manually through the appropriate Sub-CA of the Siemens Solution PKI
5. Provide the HES Client Certificates to Honeywell / Kaifa for Manufacturing

It is important for KMS to maintain a record of which of these Certificates are installed in each Meter as the Private Keys behind the certificates are used in HLS7 and Security Suite 1 operations. These Certificates cannot be statically defined as they will be included in a renewal process, so at some point in the Solution’s lifecycle there will be more than one Certificate for each role ‘in use’.

The initial link between Meter and HES Client Certificates installed could be established in a few different ways, two options are:

* In the Meter Certificate Shipment File, Honeywell / Kaifa also include the HES Client Certificates that were installed in the Meter. This would involve Honeywell / Kaifa agreeing to provision of these records in their Shipment File.
* On a per ORG basis, KMS has configuration setup that defines “for any new Meter to HES Client Certificate records created, use these Certificates”.

**Development areas are likely to be:**

* KMS - **Extend** KMS to allow import of the HES Client Certificates, in such a way that the link to the Key Alias of the private key behind the certificate is maintained, so they are visible in the KMS UI and can be included in Lifecycle Management.
* KMS - **Extend** the KMS storage model to allow for the storage of per Meter records of which HES Client Certificates are installed.
* KMS and/or KMS - **Extend** MDM and/or KMS to enable the creation of the initial link between a meter and the relevant certificates.

### HES Client Certificate Lifecycle Management

***Impact: KMS* | *MDM* | *HES***

Eventually it will be necessary to renew the HES Client Asymmetric Keys. It is probably acceptable that the monitoring of the ‘age’ of the HES Client Keys is monitored manually, and generation of a new Keypair and subsequent certificate generation will also be a manual process. The new Certificate must be installed following some of the processes described in section 2.3.4 -Meter to HES Client Certificate Mapping.

Once new HES Client Certificates are created, a process must be initiated to facilitate the installation in Meter devices of the new certificates, in bulk.

This is probably a good candidate for MOC initiation.

The possible process could be roughly:

1. MOC initiates an ‘install new HES Certificate’ Campaign
2. The Campaign results in a MOC Request being made to UDIS containing:
   1. A list of Meters to update
   2. The Certificate to install
   3. The Role the certificate is for
   4. The Certificate/Private Key ‘Alias’ for use in step #4
3. Or alternatively; MOC initiates an MDM Workflow that triggers the UDIS process
4. Or alternatively; KMS lifecycle management initiates an MDM Workflow that triggers the UDIS process
5. UDIS installs the new HES Certificate in the Meters, one by one
6. UDIS notifies KMS, per meter, as soon as the status is known, of the success / failure of the update so KMS can update the Meter -> HES Certificate relationships.

*\*\*\* If the update fails, the rollback functionality similar to that seen elsewhere in KMS for symmetric Key renewal should be implemented. \*\*\**

**Development areas are likely to be:**

* Create a **New** MDM Workflow to orchestrate the process
* MDM - **Extend** the parameter definition for the UAA UDIS interface to include a type that indicates to UDIS it must initiate an Asymmetric key renewal, the new MDM workflow will use this.
* UDIS - Create a **New** UDIS workflow to orchestrate the UDIS -> Meter interactions
* KMS – **Extend** keyUpdate Service or create a **New** Service to allow KMS to receive notification of an update status, enabling update or rollback of Meter -> HES Client records as required by *2.3.4 Meter to HES Client Certificate Mapping.*

## Security Suite 1 - HLS7 Authentication

***Impact: KMS* | *HES***

For all HES -> Meter communcations, the COSEM Association must be established using HLS7 authentication. HLS7 is a mutual authentication mechanism based on generation of a digital signature (ECDSA). Generation of a digital signature requires use of a private key. Verification of the signature requires the use of a previously distributed public key (distributed via certificate).

The process goes roughly:

1. Both Meter and HES generate and send each other a Challenge, KMS provides an existing Service to generate a Challenge of the appropriate length (rest/crypto/getChallenge)
2. Each party then concatenate each other’s Challenges, and each other’s System Titles to create a string that they generate an ECDSA signature for. These signatures are then exchanged.
3. Each party is able to construct the string that formed the input to the signature generation of the other party. This is used in conjunction with the public key of the sender and the signature they sent to verify the signature, and hence the authenticity of the sender.

HES does not have access to private or public key material, so will rely on KMS to provide functions to support the above.

**Development areas are likely to be:**

* UDIS – A **New** process that can be applied to all meter Workflows will need to be developed to replace HLS5 authentication with HLS7
* KMS – A **New** service will be required to expose the cosemGenerateSignature Worldline JSS Function for signature **generation**. This will need to take DeviceSerial, Role and a string forming an appropriate concatenation of System Titles and Challenges, as inputs.
* KMS – A **New** service will be required to allow UDIS to request signature **verification**. This will need to take DeviceSerial, Role, Signature string, and a string forming an appropriate concatenation of System Titles and Challenges, as inputs.

## Key Agreement Key Exchange Mechanism

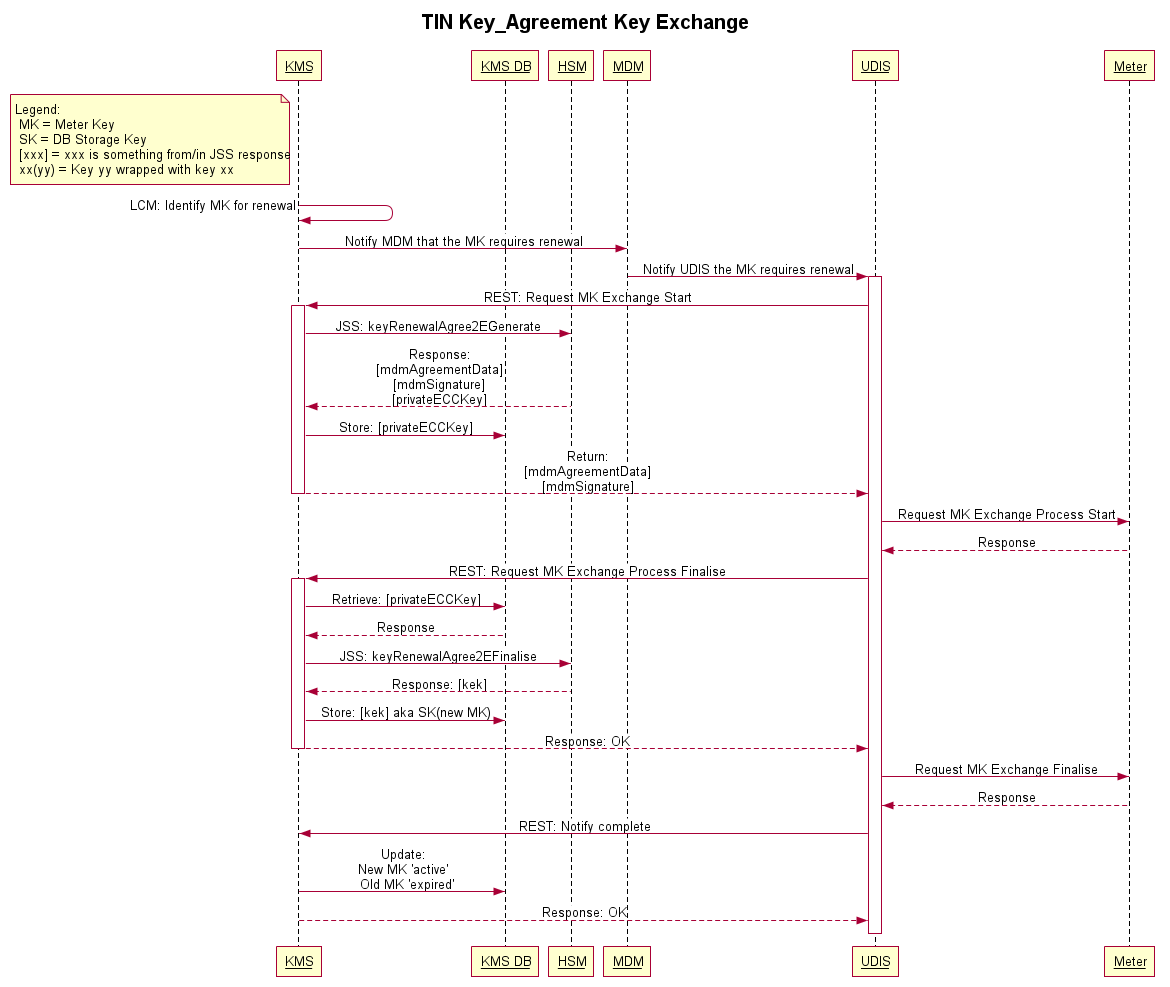
***Impact: KMS* | *HES* | *MDM***

In previous implementations, key exchange / replacement for symmetric COSEM keys has been performed using Key-Wrap mechanisms. This provides a simple end-to-end process of:

1. Generate a new key
2. Wrap the new key with the meter KEK
3. Send the wrapped new key to the meter

The Honeywell meter uses capabilities provided by Security Suite 1 to make the key renewal process more secure, using the Key Agreement mechanism *‘Ephemeral Unified Model C(2e, 0s, ECC CDH)’*. This process avoids the need to ever send the actual final key between Client and Server. In this process data is exchanged resulting in the ability of both Client and Server to compute the same Key, this Key is then stored by both Parties and becomes the new symmetric key.

The Worldline JSS provides two functions to support the process, which is broadly a 2-step process, where the output of the first is used to supplement the inputs to the second, something like:



**Development areas are likely to be:**

* HES - Creation of **New**, or **Extend** of existing UDIS Key Exchange Workflows to orchestrate the Key Agreement exchange process between HES and Meter
* KMS – **Extend** Key Templates to allow specification of Key Agreement Renewal
* KMS – **Extend** Lifecycle Management to use Key Agreement mechanisms if specified in the Key Template
* KMS – Create a **New** service to expose Worldline JSS function keyRenewalAgree2EGenerate.
* KMS – Create a **New** service to expose Worldline JSS function keyRenewalAgree2EFinalise
* KMS – **Extend** the KMS DB storage model to provide the ability to store some of the response from keyRenewalAgree2EGenerate (the resulting ephemeral key), so that it can subsequently be used in the keyRenewalAgree2EFinalise step.

## Frame Counter Handling

***Impact: KMS* | *HES* | *HSM***

Frame Counters (FC) are used to prevent against replay attacks. In simple terms, each request made to the meter contains a counter, each time a request is sent the counter must increase otherwise the meter will reject the request. Counters are maintained per COSEM Client.

Where the HES does not know the FC, there are several options to find it:

1. Estimate / Iterate upwards the FC value until a value that works is found.
2. Renew Meter Keys using a very high (and hence likely to succeed) FC value. The Key renewal results in the FC being reset for the relevant COSEM Client.
3. Retrieve the current FC from the Meter in a way where only an authorised party is able to ascertain the FC from the response.

Option #1 and #2 are not very elegant. Honeywell have proposed a manufacturer specific Solution for #3:

*“the DLMS client will invoke a method on a vendor-defined IC (get\_frame\_counter), passing in a 64 byte randomly generated challenge, which shall be generated by a FIPS 140-2 or AIS 31 compliant RNG.*

*At this point, the meter will first generate a response to the challenge by performing a HMAC-SHA256(m, K), where:*

* *m is defined as SysT-S || SysT-C || Challenge || FC where:*
  + *SysT-S: The system title of the destination of the request (the recipient, or server) : 8 bytes.*
  + *SysT-C: The system title of the source of the request (the originator, or client) : 8 bytes.*
  + *Challenge: The random challenge received in the request : 64 bytes.*
  + *FC: The frame counter to be returned : 4 bytes.*
* *K is the authentication key associated to the requested client.*

*It will then return a struct { challenge-response, frame counter }, where:*

* *challenge\_response is the result of the HMAC\_SHA256 calculation.*
* *frame\_counter is the requested frame counter.*

*The DLMS client, upon reception of the response, can then (provided the AK of the requested client is available to it) validate the challenge-response indeed originates from the meter, and additionally that the returned frame counter has not been tampered with (by performing the same calculation and checking whether the result is the same), and if so, does store the frame counter for use. Note that the DLMS client only performs this once for every meter that joins the network (and not for every session).”*

There is currently no provision in HES, KMS or HSM JSS for the required functions to support option #3 solution.

**Development areas are likely to be:**

* HSM - Creation of **New** JSS function to expose a capability to use Symmetric Meter Keys to verify HMACs
* KMS – Create a **New** service to expose Worldline JSS function described above
* HES – **Extend** the Honeywell Auto-Registration workflow to enable retrieval of a subsequent storage and use of the Frame Counter