

KULLANICI KILAVUZU (UYGULAMA İÇİ, TAM)

OPC UA Server

Çevrimiçi: mergenhub.envest.com.tr/tr/docs/kilavuz-opcua

BU SAYFADA

Endpoint & Bağlantı

Güvenlik & Sertifikalar

_System Tag'leri

Desteklenen Servisler (profil)

Tag Bazlı Yetki (ACL)

Geçmiş Okuma (Historical Access)

Sunucu Metotları (Call)

Alarm & Events (A&E) Yayını

Endpoint & Bağlantı

OPC UA Server, tüm tag'leri `opc.tcp://host:48010/MergenHub` adresinde sunar (Ayarlar → OPC UA).

Host

`localhost` (yalnız yerel) veya `0.0.0.0` /makine adı (LAN).

Port / EndpointPath

Varsayılan 48010 / MergenHub.

Max. connections / session timeout

proje özellikleri paritesi (çok-oturumlu istemciler için uzun timeout).

Güvenlik & Sertifikalar

Endpoint güvenlik policy'leri seçilebilir: **None**, **Basic256Sha256** (Sign / Sign&Encrypt) vb. Anonim veya kullanıcı adı/şifre token'ı. Güvenilmeyen istemci sertifikaları "rejected" listesine düşer; **Ayarlar** → **Certificates**'tan "trusted" yapılır. Sunucunun kendi instance sertifikası ilk açılışta üretilir.

_System Tag'leri

Cihaz ve sunucu istatistikleri `_System` / `_Statistics` tag'leri olarak OPC'de sunulur:`_System._ActiveTagCount`

Aktif (abone/taranan) tag sayısı.

`<cihaz>._System._DemandPoll`

Yazınca cihazı bir kez zorla taratır (write-triggered poll).

`<cihaz>._System._Enabled`

Cihaz taraması aç/kapa (yaz).

`._Statistics._SuccessfulReads /
_FailedReads / _SuccessfulWrites`

Sayaçlar.

`._Statistics._Error`

Cihaz hata durumu (0/1).

`<cihaz>._System._Error /
_NoError`Cihazın **haberleşme sağlığı**. Yalnız "son işlem hatası" değil, **gerçek veri akışı** da hesaba katılır: bağlantı açık ama veri gelmiyorsa (zombi oturum) `_NoError=0` olur.

İpucu Bir istasyonun sağlığını izlemek için `_NoError` 'a abone olmak yeterlidir — abonelik cihaz için tarama talebi oluşturur, cihaz gerçekten yoklanır. Devre dışı ya da hiç tüketicisi olmayan (bilerek boşta) cihaz hata sayılmaz.

Not Bu sistem tag'leri de aboneliğe/monitored-item'a dahil olduğundan, OPC panelindeki abone/item sayısı ham cihaz tag sayısından birkaç fazla görünür — normaldir.

Desteklenen Servisler (profil)

MergenHub bir **Standard UA Server**'dir: **Data Access + Historical Access** (ham & aggregate) + **Method Call**. Aşağıdaki tablo neyin sunulduğunu ve neyin sunulmadığını gösterir.

Browse / BrowseNext / TranslateBrowsePath	Var — yetkiye göre süzülür.
Read / Write	Var — değer canlı TagStore'dan; yazma cihaz kuyruğuna gider.
Subscribe (MonitoredItem / Publish)	Var — veri değişimi push edilir.
HistoryRead	Var — Raw + Processed (Average/Minimum/Maximum/Count). Bkz. "Geçmiş Okuma".
Call (metot)	Var — <code>_Methods</code> klasörü. Bkz. "Sunucu Metotları".
HistoryUpdate (geçmişe yaz/sil)	Yok — geçmiş salt-okunurdur.
OPC UA Alarms & Conditions (event modeli)	Var — <code>AlarmConditionType</code> koşulları. Bkz. "Alarm & Events (A&E) Yayını".
OPC UA PubSub (UADP/JSON)	Yok — bulut/mesajlaşma için Veri Köprüsü (MQTT/REST).
AddNodes / DeleteNodes, Query	Yok — adres uzayı config'ten üretilir (REST Configuration API).
LDS'e kayıt, X509 kullanıcı token'i	Yok — doğrudan endpoint URL'i; UserName/Password + kanal sertifikası.

Ayrıntı Tam profil belgesi: `docs/opc-ua-server.md` .

Tag Bazlı Yetki (ACL)

Ayarlar → **OPC UA** → **Kullanıcılar** → **Yetki**'den kullanıcı (ve anonim oturum) başına kanal/cihaz/tag yolu üzerinde erişim yok · okuma · okuma-yazma verilir. Çözümleme **en uzun örnek kazanır**.

Browse	Yetkisiz düğüm hiç listelenmez (erişilebilir tag'in ata klasörleri görünür kalır).
Read / Write / Abonelik	<code>BadUserAccessDenied</code> .
HistoryRead	<code>BadUserAccessDenied</code> — okuyamadığı tag'in geçmişini de okuyamaz .
Call (metot)	Hedef üzerinde okuma-yazma yoksa <code>BadUserAccessDenied</code> .

Hiçbir kullanıcıya yetki atanmamışsa kısıt yoktur (geriye uyum). Her red denetim günlüğüne düşer (spam koruması ile).

Geçmiş Okuma (Historical Access)

OPC UA istemcileri **geçmiş veriyi** doğrudan sunucudan okuyabilir. Kaynak **Data Logger**'dir — OPC için ayrı bir depolama kurulmaz.

Hangi tag?	Yalnız bir Data Logger grubuna eklenmiş tag'ler. Onların node'unda <code>Historizing = true</code> ve <code>HistoryRead</code> erişim biti bulunur; diğerleri <code>BadHistoryOperationUnsupported</code> döner.
Ham (raw)	<code>StartTime / EndTime (bitiş hariç)</code> , <code>NumValuesPerNode</code> , <code>ReturnBounds</code> . <code>StartTime > EndTime</code> → yeniden eskiye.
Aggregate	Average · Minimum · Maximum · Count . Kova zamanı kova başlangıcıdır; boş kovada Count 0, diğerleri <code>BadNoData</code> .
Sayfalama	Çağrı başına değer sınırı Ayarlar → OPC UA 'dan (varsayılan 5000). Sınıra ulaşırsa devam noktası döner; istemci kalanı onunla alır.
Kalite / zaman	Kalite Good/Uncertain/Bad olarak; kaynak zaman damgası sürücünden gelen örnek zamanıdır.
Desteklenmeyen	<code>ReadAtTime</code> , <code>ReadModified</code> , event geçmiş, <code>HistoryUpdate</code> (geçmişe yazma/silme).

UaExpert ile: düğümü seçin (`Historizing` true olmalı) → **Document** → **Add** → **History Trend View** → aralığı seçin. Aggregate için görünümün **Configuration** bölümünden `Average` + `Processing Interval` girin.

Kapatma Ayarlar → **OPC UA** → **Geçmiş okuma** kapatılırsa bayraklar hiç konmaz ve `HistoryRead` desteklenmez (özellik öncesi davranış).

Ayrıca bkz [Data Logger](#)

Sunucu Metotları (Call)

Sunucu kökündeki `_Methods` klasöründe çağrılabilir metotlar bulunur. Metotlar cihaz/tag başına çoğaltılmaz — hedef, **argüman** olarak verilir (büyük projelerde adres uzayı maliyeti sabit kalır).

<code>ResetTotalizer(TagPath)</code>	Totalizer gelişmiş tag'inin birikimini sıfırlar (<code>_AdvancedTags.Ad</code> ya da yalnız <code>Ad</code>).
<code>AcknowledgeAlarm(AlarmId)</code>	Aktif alarmı onaylar — REST <code>alarms/active/{id}/ack</code> ile aynı işlem.
<code>ForceScan(DeviceKey)</code>	Cihazı zorla taratır — <code>._System._DemandPoll</code> yazmakla aynı (<code>Kanal.Cihaz</code>).
<code>ResetStatistics(DeviceKey)</code>	<code>_Statistics</code> sayaçlarını sıfırlar.

Hepsi (**Succeeded: Boolean, Message: String**) döner. İş sonucu başarısızlığı protokol hatası değildir: çağrı `Good` döner, neden `Message` 'da yazar.

Yetki + denetim Her çağrı, argümandan çözülen hedef üzerinde **okuma-yazma** yetkisi ister (yoksa `BadUserAccessDenied`) ve başarılı/başarısız fark etmeksizin **denetim günlüğüne** düşer. Hedef çözülemezse çağrı `BadInvalidArgument` alır — yetki atlanamaz.

Bağımlılığı olmayan metot (ör. alarm modülü yoksa `AcknowledgeAlarm`) **adres uzayına hiç konmaz** — `BadNotImplemented` dönen hayalet düğüm bulunmaz. **Ayarlar** → **OPC UA** → **Sunucu metotları** kapatılırsa

_Methods klasörü de yaratılmaz.

Ayrıca bkz [Tag Bazlı Yetki Alarmlar](#)

Alarm & Events (A&E) Yayını

MergenHub'ın ISA-18.2 alarm motoru OPC UA'da AlarmConditionType koşulları olarak yayınlanır. **İkinci bir alarm sistemi yoktur**: tanım, durum ve onay tek kaynaktan (Alarmlar modülü) gelir — web arayüzünden yapılan onay UA abonelerine, UA'dan yapılan onay web arayüzüne yansır.

Ayar	OpcUa.PublishAlarms (varsayılan açık). Kapatılırsa _Alarms düğümleri hiç kurulmaz, adres uzayı eski hâliyle kalır.
Adres uzayı	Objects → _Alarms → [Alan] → <alarm> . Namespace: http://envest/mergenhub/ua/alarms .
Abonelik	İstemci Server düğümüne olay aboneliği açar (_Alarms Server'a HasNotifier ile bağlıdır). Dilerse yalnız bir Alan klasörüne abone olur.
ConditionRefresh	Bağlanır bağlanmaz çağrılır → aktif/onay bekleyen alarmların anlık listesi gelir.
Onay	AcknowledgeableConditionType.Acknowledge → mevcut ISA-18.2 onayı; denetim günlüğüne düşer.

ISA-18.2 ↔ UA durum eşlemesi:

Durum	ActiveState	AckedState	Retain
Normal	false	true	false
Alarm (onaysız)	true	false	true
Alarm (onaylı)	true	true	true
Normale döndü, onaysız	false	false	true

Koşul tipi ConditionName alanında taşınır: HIHI / HI / LO / LOLO / ROC_HI / ROC_LO / DEV_HI / DEV_LO . Severity motorla **aynı 1-1000** ölçeğindedir. SourceName tag yolu, SourceNode ise gerçek tag düğümüdür. Veri kesilmişse (BAYAT alarm) Quality = UncertainLastUsableValue olur.

Yetkilendirme A&E de **tag yetkisine tabidir**: kaynak tag'i **okuyamayan** kullanıcıya o alarm hiç yayınlanmaz (abonelik ve ConditionRefresh süzülür, düğüm browse'da görünmez); **onay** için okuma-**yazma** gerekir, yoksa BadUserAccessDenied . Yetki değişikliği açık oturumlara **canlı** yansır.

Sınırlar Tek olay tipi (AlarmConditionType) sunulur; branch/Confirm/Shelving desteklenmez; olay geçmişi UA HistoryRead ile değil GET /alarms/v1/history ile okunur. Durum değişimleri 500 ms çözünürlükle örneklenir.

Ayrıca bkz [Alarmlar \(A&E\)](#), [Alarmların OPC UA'dan Okunması](#)