



# OpenOpcUa

## implémentation OPC UA Freeware



Looking for North-America partner



**Michel Condemine**

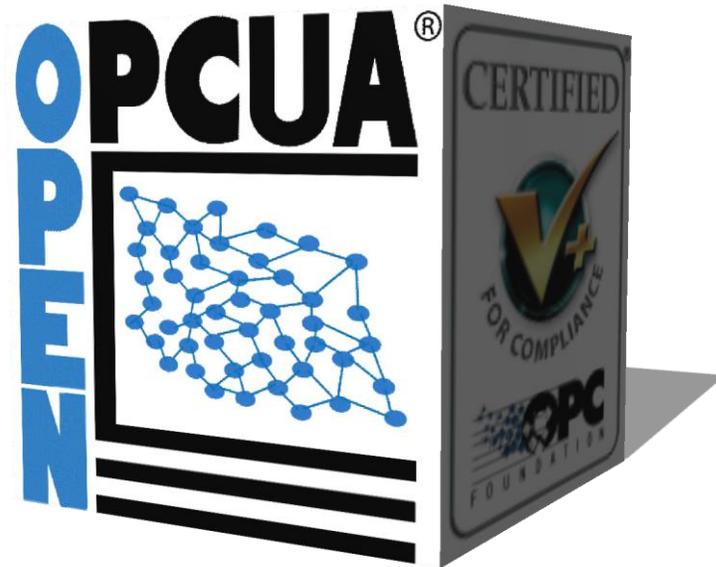
- OPC Foundation France (Technique)
- Directeur de **4CE Industry**
- Leader du projet OpenOpcUa
- Membre de l'OPC Foundation TAC
- MichelC@4CE-Industry.com





# Nouveautés 2024

## Résumé des nouveautés





# Nouveautés 2024

- Mode de licence
  - OpenOpcUa (OOUA) n'est plus en OpenSource,
  - OOUA est un freeware
- Le code source des squelettes de VPIs et VFIs reste Open Source
- Bug fixes :

• OOUACoreServer	28
• OOUAStack	7
• OOUASharedLib	1
• OOUAClientLib	9
• XiMulator	4
• OOUAConfigManager	17
• OOUANodeManager	0
- New features

• OOUACoreServer	30
• OOUAStack	6
• OOUASharedLib	1
• OOUAClientLib	1
• XiMulator	15
• OOUAConfigManager	+20
• OOUANodeManager	3



# Nouveautés 2024

- VPIs
  - **Vpi Library ver: 1.0.5.1**
  - VpiNullEx 1.0.0.4
  - VpiUaClient ver: 1.0.6.4
  - VpiEthIp 1.0.5.8
  - VpiS7 ver: 1.0.3.6
  - VpiUaMqttJsonPub ver: 0.0.1.6
  - VpiUaMqttJsonSub ver: 0.0.0.8
  - VpiMdbMaster 1.0.4.6
  - VpiMdbServer 0.0.0.6
  - VpiValueInit 0.0.3.5



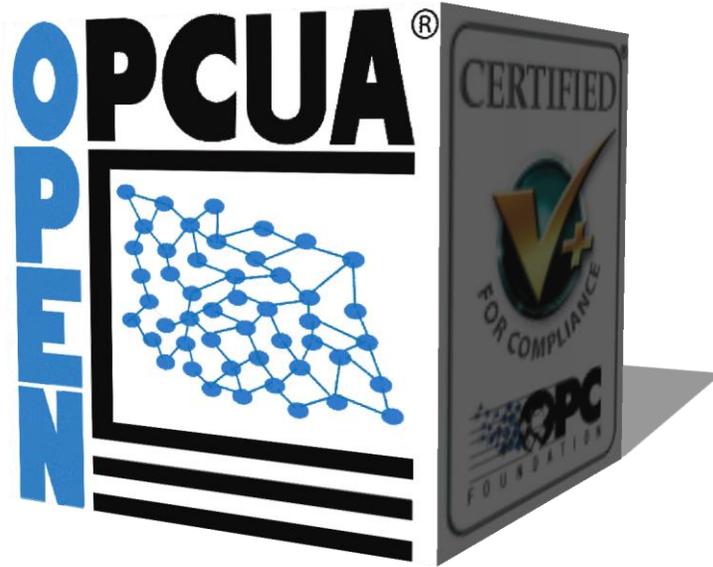
# Nouveautés 2024

- VFIs
  - **VFI Library ver:** **0.0.1.1**
  - VfiPgSql v0.0.7.5
  - VfiCsv RAS
  - VfiSqlServer RAS
  - VfiParquet Project 2025



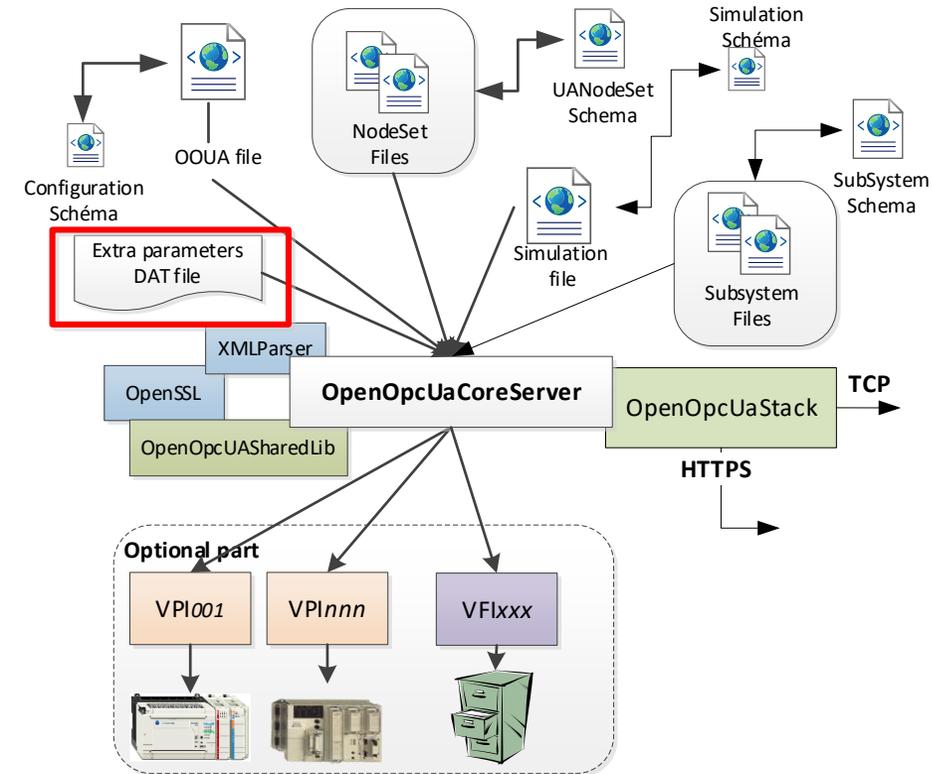
# Nouveautés 2024

OPCUA CoreServer



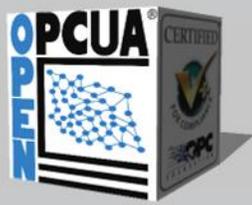


# Nouveautés 2024



## Fichier de configuration complémentaire

- LOG\_MAX\_ENTRIES Spécifie le nombre max de lignes de log dans chaque fichier
- MAX\_BACKUP\_FILES Spécifie le nombre de fichiers à conserver.
- SAVE\_SYSTEM\_NODES
- HDC\_URI
- HDC\_NODEID
- MODEL\_CHANGE\_SUPPORT
- FILE\_BUFFERIZATION



# Nouveautés 2024

Une nouvelle fonction callback est disponible pour les VFIs. Elle permet de récupérer la description de nœuds, notamment DisplayName, BrowseName, References, etc.

Elle nécessite, côté VFI, l'implémentation de **VfiSetNodesDescriptionCallback** avec la signature suivante :

**Vfi\_StatusCode VfiSetNodesDescriptionCallback(Vfi\_Handle hVfi, PFUNCVFINODESDESCRIPTIONCALLBACK lpVfiNodeDescriptionCallback);**

La signature du callback elle-même est la suivante :

```
typedef Vfi_StatusCode(__stdcall *PFUNCVFINODESDESCRIPTIONCALLBACK)(Vfi_UInt32 uiNoOfNotifiedObject, Vfi_NodeId* Id, OOUABrowseDescription* pBrowseDescription);
```

## Structures associées

```
typedef struct _OOUABrowseDescription  
{
```

```
    Vfi_NodeId          m_NodeId;  
    Vfi_NodeClass m_NodeClass;  
    Vfi_QualifiedName* m_pBrowseName;  
    Vfi_LocalizedText m_DisplayName;  
    Vfi_LocalizedText m_Description;  
    Vfi_UInt32         m_WriteMask;  
    Vfi_UInt32         m_UserWriteMask;  
    Vfi_UInt32         m_nbOfReferences;  
    Vfi_ReferenceNode* m_pReferences;  
    Vfi_Boolean        m_bAbstract;  
    Vfi_Boolean        m_bSymetric;  
    Vfi_LocalizedText m_InverseName;  
    Vfi_String         m_ReleaseStatus;
```

```
typedef struct _Vfi_ReferenceNode
```

```
{  
    Vfi_NodeId ReferenceTypeId;  
    Vfi_Boolean IsInverse;  
    Vfi_ExpandedNodeId TargetId;  
}  
_ReferenceNode;
```



# Nouveautés 2024

## Comportement du serveur

- Le serveur, à partir de la version 1.1.5.7, accepte désormais une discordance entre la déclaration de l'accessLevel UAVariable et la déclaration des AccessRight des tags au sein des VPIs.
- Un nouveau mécanisme de mise en mémoire tampon, hors ligne, des écritures VFI est géré par le serveur. Les données de stockage hors ligne sont conservées dans un fichier ASCII appelé **<ARCHIVE-NAME>-PendingRequest.dat**. Ces données sont restaurées au redémarrage du serveur dans plusieurs scénarios, panne du serveur, déconnexion de la base de données, etc. Le mécanisme de mise en mémoire tampon vers un fichier externe peut être désactivé à l'aide du fichier DAT du serveur. Utilisez le paramètre FILE\_BUFFERING\_MECHANISM avec la valeur True ou False pour activer le comportement approprié. La valeur par défaut est False.

## Support des Events pour les VPIs

- Une nouvelle Callback est disponible pour les VPIs. Elle permet aux VPIs d'envoyer des événements définis par l'utilisateur.

## Changement dans le schéma ServerConfig.xsd

- Un nouvel attribut optionnel est disponible dans ServerConfig.xsd. Cet attribut s'appelle **Disable** et s'applique aux éléments XML FileNodeSet et FileSubsystem. Il permet de désactiver un fichier dans la configuration.



# Nouveautés 2024

## Archivage de nœuds systèmes

Le serveur enregistre désormais certains nœuds système via le VFI configuré.

Ces nœuds comprennent :

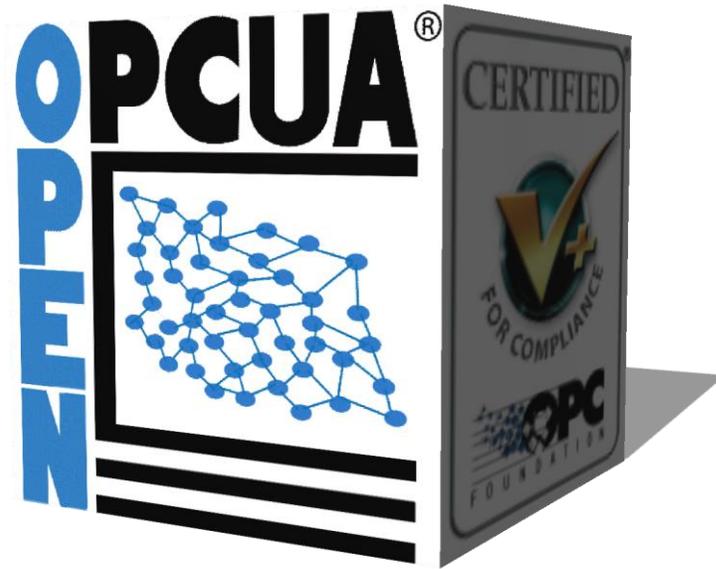
- 2 nœuds OPC Foundation (StartTime, State),
- 2 nœuds par VPI (DeviceState, ActiveState),
- 3 nœuds par VFI (HaEngineState, VfiActiveState, HAEngineStatus)

Cette fonctionnalité est optionnelle et s'active dans le fichier de configuration complémentaire en utilisant le paramètre `SAVE_SYSTEM_NODES`



# Nouveautés 2024

00UASack





# Nouveautés 2024

## Nouvelles fonctions dans la Stack

`OpcUa_DateTime_AddDuration(OpcUa_DateTime* pdtVal, OpcUa_UInt64 duration)`

Ajoute duration à pdtVal

`OpcUa_DateTime_CompareEx(OpcUa_DateTime dt1, OpcUa_DateTime dt2)`

compare dt1 and dt2.

renvoi -1 si  $dt1 < dt2$

renvoi 0 si  $dt1 == dt2$

renvoi 1 si  $dt1 > dt2$

`OpcUa_DateTime_Sub(OpcUa_DateTime dt1, OpcUa_DateTime dt2, OpcUa_Double* pdblDiff)`

Calcul la différence entre dt et dt.

Le calcul n'est réalisé que si  $dt1 > dt2$  sinon la fonction retourne -1.

## Evolutions

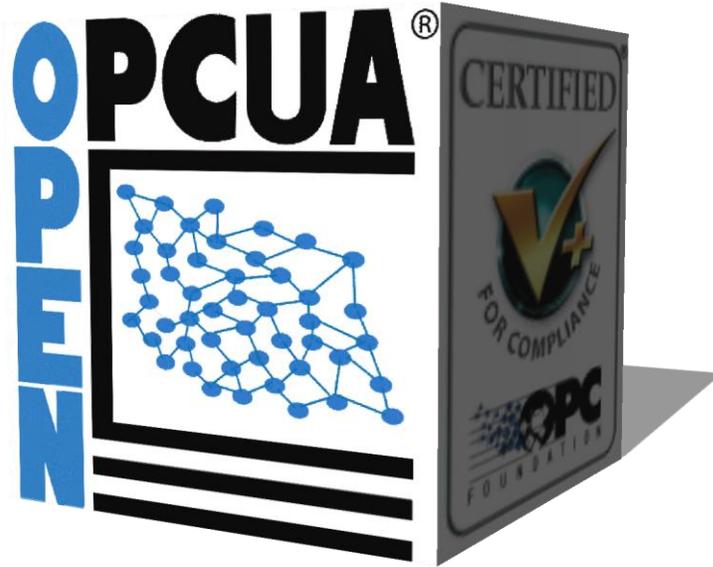
Changement du comportement de `OpcUa_Thread_Delete` et de `OpcUa_P_Thread_Delete`, Chacune d'entre elles attend 120 secs au lieu de « TOUJOURS » avant de s'auto-détruire.

Modification de `MAX_SECURECHANNELS` et `MAX_SESSIONS`. `MAX_SECURECHANNELS = MAX_SESSIONS + 1` et `MAX_SESSIONS = 50` comme requis par le profil standard.



# Nouveautés 2024

OOUASharedLib



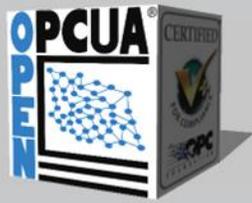


# Nouveautés 2024

## Nouveautés dans l' OOUASharedLib

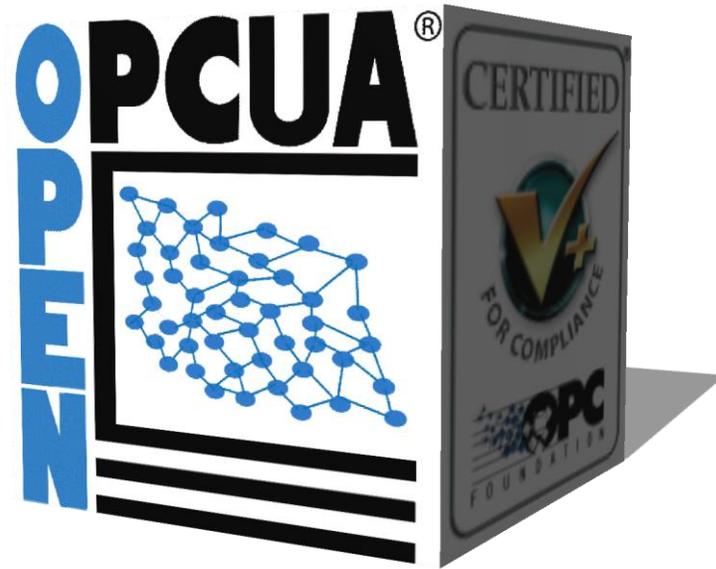
La fonction **NodeId2String** pour NodeId pour prendre en charge la conversion des nœuds au format **GUID**.

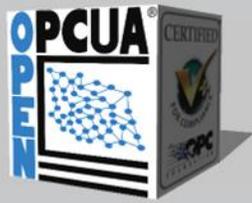
Correction d'une fuite mémoire dans la méthode CDataValue::TransfertStringArray



# Nouveautés 2024

OOUAClientLib





# Nouveautés 2024

## Nouveautés dans la OOUAClientLib

### Evolution

- L'arrêt de la WatchingThread attendra 120 secondes au lieu de 10 secondes avant de détruire la session.
- Modification du mécanisme d'arrêt de la surveillance associé à la WatchingThread
- Introduction d'une méthode **IsStopping** pour la consolidation globale des séquences d'arrêt.
- Suppression des variables globales.
- Au démarrage, une publication inutile est envoyée par la bibliothèque même si les sessions sont désactivées.
- ClientLib n'accusait pas réception de la notification. Ce problème est maintenant résolu.  
Il s'agissait d'une régression introduite dans la version 1.1.0.7
- Modification du mécanisme de reconnexion.  
À partir de la version 1.1.1.3, la boucle de récréation est désormais interrompue en cas d'échec.

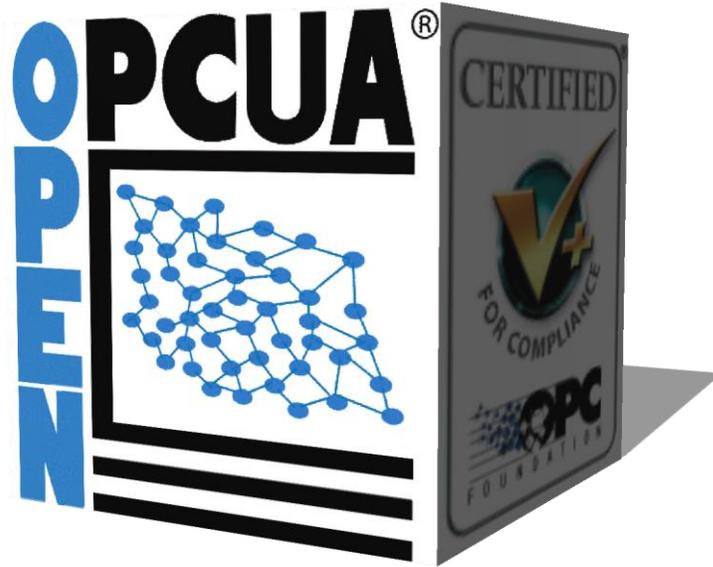
### Correction

- Correction du problème d'initialisation lors de l'appel du callback de l'hôte
- Correction du problème d'initialisation sur HANDLER\_CLIENTCONFIG\_DATA
- Lorsque qu'aucune Notification n'était disponible, les changements de Status du serveur n'était pas envoyé.



# Nouveautés 2024

OOUAVpiLibrary





# Nouveautés 2024

## Nouveautés dans le GenericEncoderDecoder

- Nouvelle fonctionnalité de journal
- Ajout du codage/décodage NodeId
- Ajout du décodage pour les types Float, UInt64 et Double.

## Correctif dans le GenericEncoderDecoder

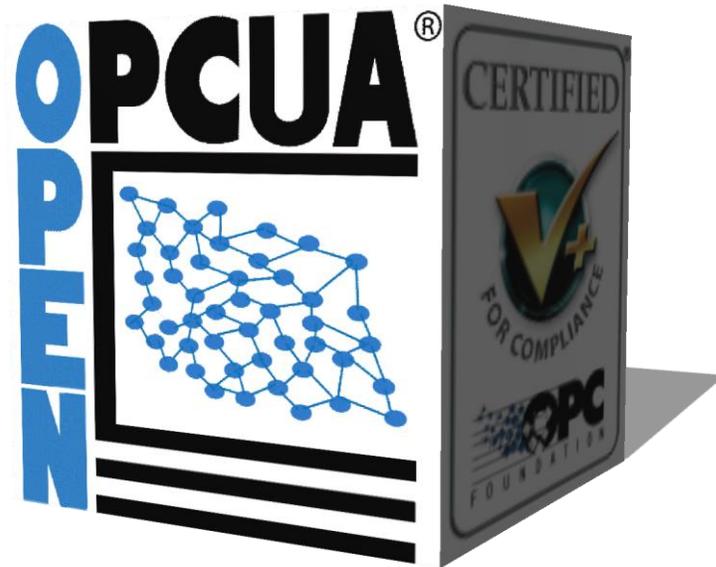
Prise en charge de la définition de plusieurs dictionnaires  
Renvoie une erreur lorsqu'aucun dictionnaire n'est disponible.





# Nouveautés 2024

VPIs





# Nouveautés 2024

## Nouveautés dans l'OOUAVpiLibrary

Ajout d'une durée (dtVal) fournie en millisecondes à un Date\_Time (pIntOutDate) passé en paramètre.

```
Vpi_StatusCode Vpi_DateTime_AddDuration(Vpi_UInt32 dtVal, Vpi_DateTime* pIntOutDate)
```

API pour la prise en charge la persistance NodeId dans les VPIs

```
Vpi_StatusCode Vpi_NodeId_InitializePersistantLayer(PersistantDictionary** ppPersistantDictionary);
```

```
Vpi_StatusCode Vpi_NodeId_LoadPersistantLayer(const Vpi_String szFullFileName, PersistantDictionary** ppPersistantDictionary);
```

```
Vpi_StatusCode Vpi_NodeId_ReleasePersistantLayer(PersistantDictionary* pPersistantDictionary);
```

```
Vpi_StatusCode Vpi_NodeId_MakePersistant(PersistantDictionary* pPersistantDictionary);
```

Structure associée

```
typedef struct _PersistantDictionary
```

```
{
```

```
    Vpi_CharA* pszFullFileName;
```

```
    std::map<std::string, Vpi_NodeId>* pInternalPersistantDictionary;
```

```
} PersistantDictionary;
```



# Nouveautés 2024

## Nouveautés dans l'OOUAVpiLibrary

- Fonction permettant de convertir un StatusCode en un message compréhensible par l'utilisateur. Fondamentalement, il convertit le StatusCode en chaîne  
**Vpi\_StatusCode Vpi\_String\_FromStatusCode(const Vpi\_StatusCode uInputStatus, Vpi\_String\* pszStatusCode);**
- **Vpi\_Trace** gère désormais automatiquement la taille du fichier Log, en nombre de lignes. Il clonera le dernier fichier vers une copie appelée : <SubSystemName>MM-YY-hh-mm-ss>.bak  
Pas plus de uiMaxBackup ne seront créées. La bibliothèque conservera le dernier fichier dans le dossier log.  
**Vpi\_Void Vpi\_Trace\_SetOutFileNoOfEntriesMax(void\* pProxyStubConfiguration, Vpi\_UInt32 iOutFileNoOfEntries);**  
il est censé être utilisé à partir du code Vpi appelant la nouvelle fonction Vpi optionnelle VpiSetLogMaxEntries.  
**Vpi\_StatusCode VpiSetLogMaxEntries(Vpi\_Handle hVpi, Vpi\_UInt32 uiLogMax, Vpi\_UInt32 uiMaxBackup);**

## Correctif dans l'OOUAVpiLibrary

- Modification de la méthode get/set de CRtSocket de *GetPort* à **GetTcpPort** et de *SetPort* à **SetTcpPort** pour la conformité avec Microsoft WinSocket.
- Ajout de tests supplémentaires sur le nom de fichier dans la fonction Vpi\_Trace\_Internal pour la sécurisation.
- Optimisation de la Mutex de la fonction de Trace
- Le format des messages de LOG sont modifiés. Ils sont maintenant harmonisés avec le format utilisé par le serveur et Vpi.



# Nouveautés 2024

## Nouveautés dans le VpiS7

- Modification de la séquence d'initialisation.  
Si la communication n'est pas possible depuis ColdStart, le VpiS7Thread démarrera et le WarmStart sera appelé.
- Un nouveau message est ajouté au niveau info pour informer les utilisateurs lorsque la communication redémarre.

## Correctif dans le VpiS7

- Correction d'un problème sur la notification de changement de valeur
- La version VpiS7 <1.0.3.5 faisait face à une double erreur de libération en cas d'erreur d'analyse. Ceci est maintenant corrigé dans la version 1.0.3.6 de VpiS7.



# Nouveautés 2024

## Nouveautés dans le VpiUaClient

- Modification de la fonction VpiWarmStart.  
Le Status WarmStartNeed est retourné lorsque pSessionConfiguration est Nulled.
- En mode automatique, les nœuds de l'espace de noms 1 n'étaient pas agrégés.  
Ce problème est résolu dans la version 1.0.6.3
- En mode automatique, le Vpi utilise la nouvelle fonctionnalité de la VpiLibrary afin de maintenir les NodeIds inchangés

## Correctif dans le VpiUaClient

- Correction d'une fuite lors de l'appel GetSessions
- Au démarrage, la valeur par défaut n'était pas correctement initialisée. Le problème est désormais résolu
- Correction en cas de détection d'un Vpi\_BadTimeout dans la gestion du Status interne du Vpi.
- Correction du mode AutoConfiguration.  
La notification ne fonctionnait plus dans la version 1.0.6.0 quand aucun type de données n'était défini.



# Nouveautés 2024

## Nouveautés dans le VpiValuesInit

- Nouveau paramètre dans le fichier DAT appelé DATABASE\_NAME. Il permet d'avoir un nom de schéma et de base de données distinct. Ce paramètre est facultatif. S'il n'est pas spécifié, les noms du schéma et de la base de données seront identiques.
- Nouveau format pour la taille du tableau dans la charge utile
- Nouveau mécanisme d'initialisation. Le ColdStart déverrouillera le VpiValuesInitThread même si la connexion à la base de données a échoué. Il sera géré par le WarmStart de manière régulière.
- Si aucune valeur n'a été extraite de la base de données, le DeviceState est défini sur Good. Dans les versions précédentes, il restait sur NotConnected
- Prise en charge des tableaux vides
- Implémentation de la fonction VpiSetLogMaxEntries à partir de la version 0.0.3.5
- À partir de la version 0.0.3.5, le nœud DeviceState est défini sur Good uniquement une fois que des valeurs ont été transférées vers le cache du serveur.



# Nouveautés 2024

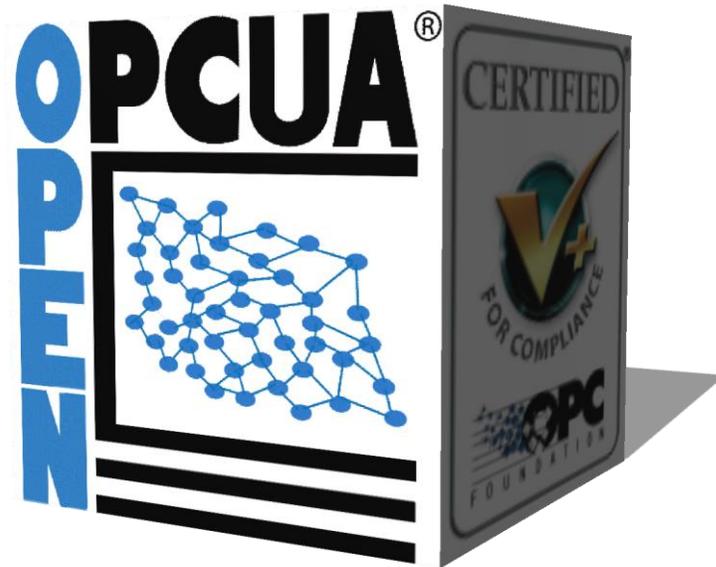
## Correctif dans le VpiValuesInit

- Correction dans la gestion de TracelLevel
- Le ValueInit peut faire une confusion entre un tableau à un élément et un scalaire. Cette situation est corrigée dans la version 0.0.2.7
- si la table n'existe pas, l'initialisation des valeurs est arrêtée
- Correction du blocage dans VpiColdStart et VpiWarmStart lors de l'utilisation de la fonction de redondance et du ValueInit.
- Afin d'améliorer les performances. CVpiValuesInit::NamespaceArrayUpdate() est désormais appelé depuis VpiValuesInitThread une fois pour toutes.
- Lorsque rien ne doit être injecté dans le cache du serveur, le DeviceState était incorrectement défini sur Good.



# Nouveautés 2024

VFI





# Nouveautés 2024

## Nouveautés dans le OOUAVfiLibrary

- Une nouvelle fonction permettant le parsing de nodeID a été introduite.

**Vfi\_StatusCode Vfi\_NodeId\_Parse(const char\* szNodeId, Vfi\_NodeId\* pNodeId)**

## Correctif dans le OOUAVfiLibrary

- Le mécanisme de trace des VFIs est désormais harmonisé avec la trace des VPIs et le trace du serveur.

## VFIs API

Une nouvelle fonction est disponible pour l'implémentation. Cette fonction s'appelle VfiHistoryWriteEx. Elle permet des opérations d'écriture en masse. Les développeurs de VFIs sont encouragés à l'utiliser.

Signature :

**Vfi\_StatusCode VfiHistoryWriteEx(Vfi\_Handle hVfi, Vfi\_UInt32 uiNbOfValueToWrite, HistoryWrite\* pValues);**

Structure :

```
typedef struct HistoryWrite
{
    Vfi_NodeId      Id;
    Vfi_UInt32      uiNbOfValue;
    Vfi_DataValue* pValues;
} _HistoryWrite;
```



# Nouveautés 2024

## Nouveautés dans le VfiPgSql

- Un nouveau mécanisme de mise en mémoire tampon hors ligne est géré par le serveur. Ce tampon hors ligne a étendu le mécanisme en ligne existant qui est géré par le VFI.
- Les données de stockage hors ligne sont conservées dans un fichier ASCII appelé **<ARCHIVE-NAME>-PendingRequest.dat**. Ces données sont restaurées dans plusieurs scénarios, panne du serveur, déconnexion de la base de données, etc.
- Le Vfi ignore les enregistrements avec `StatusCode == BadWaitingForInitialData`
- Pour une demande à plusieurs nœuds, le Vfi extrait des échantillons de la base de données dans un seul appel SELECT.
- Mise à jour dynamique des `NamespaceArray`. Ceci est utilisé lorsqu'un Vpi crée dynamiquement des nœuds/balises.
- correction dans l'extraction UDT
- Nouveau message lorsque le même enregistrement apparaît dans la liste des enregistrements en attente.



# Nouveautés 2024

## Nouveautés dans le VfiPgSql

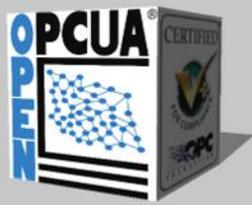
- Le VfiPgSql utilise VfiHistoryWriteEx. Cette fonction permet l'écriture en masse. Le Vfi écrira autant d'enregistrements que possible dans la collection d'enregistrements. Cela signifie que le tampon d'écriture doit être suffisamment grand. Un paramètre dans le fichier DAT permet de contrôler cette taille. Ce paramètre est appelé MAX\_BUFFER\_SIZE. Cette taille est en octets, la valeur par défaut est 150.000.000
- Un paramètre permet de contrôler le taux d'écriture dans la base de données. Ce paramètre est appelé POPULATE\_SQLSERVER\_INTERVAL, la valeur par défaut est 1000.
- Nouveau paramètre dans le fichier DAT appelé DATABASE\_NAME. Il permet d'avoir un nom de schéma et de base de données distinct. Ce paramètre est facultatif. S'il n'est pas spécifié, le nom du schéma sera utilisé.



# Nouveautés 2024

## Nouveautés dans le VfiPgsqI

- Marquer l'enregistrement comme enregistré dans n'importe quel scénario
- Écriture multiple pour les dates au format TEXT
- Prise en charge de MasterTable. La MasterTable est une table PostGres créée par le VFI dans le BD associé au VfiPgsqI. La colonne sert à créer un contexte sur la DataTable. Cette table contient la colonne suivante : Uri, Id, BrowsePath, BrowseName, DisplayName, Description, Datatype, Typeld, accessLevel, ValueRank, ArrayDimension.
- La prise en charge de la table Master doit être activée à partir du fichier DAT en utilisant le paramètre MASTERTABLE\_SUPPORT, les valeurs possibles sont TRUE ou FALSE. La valeur par défaut est FALSE.
- Nouveau format pour la taille du tableau dans la charge utile
- Le Vfi contenait 2 versions de la méthode CountChar. J'ai harmonisé le code dans CVfiInternalData::CountChar et CVfiPgsqI::CountChar.
- Suppression du paramètre FILE\_BUFFERING\_MECHANISM



# Nouveautés 2024

## Correctif dans le VfiPgSql

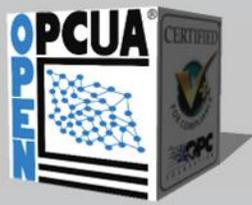
- Si l'URI d'un UDT ne peut pas être trouvé, l'enregistrement ne sera pas stocké. Cela évite les problèmes lors du rechargement et améliore la stabilité.
- Les modifications peuvent être restaurées dans la boucle suivante.
- Correction d'un problème sur le calcul de la taille d'un enregistrement. Un mauvais calcul pouvait entraîner un crash du serveur,
- Nettoyage mineur dans le mécanisme de sauvegarde, y compris un calcul de longueur d'enregistrement utilisé dans l'archivage en masse.
- VfiHistoryWriteEx renvoyait un StatusCode incorrect si `pValues[i].uiNbOfValue==0`.
- Dans certains scénarios, certaines valeurs ne sont pas enregistrées par VfiPgSql. Ce problème critique peut entraîner des pertes de données. Le problème est désormais résolu avec la version 0.0.3.9.
- Un problème apparaissait lors du stockage d'entiers signés de 8 et 16 bits. La version précédente ignorait les modifications et stockait la mauvaise valeur.
- Le nombre d'éléments renvoyés sur le nœud de datatype Array n'était pas correct. Le dernier élément manquait toujours.



# Nouveautés 2024

## Correctif dans le VfiPgSql

- Correction d'un blocage dans la fonction VfiColdStart
- Correction de la mise à jour LastWrite et LastRead effectuée par le Vfi dans le cache du serveur
- Correction du mécanisme de reconnexion
- Consolidation du code lors de la tentative d'enregistrement d'un paramètre plus grand que celui configuré.
- Correction d'un problème dans la méthode CVfiPgSql::GetStringAsDate. L'horodatage a été converti en une date NULL en millisecondes quand la valeur à convertir est hors de portée.



# Nouveautés 2024

## Nouveautés dans le XiMulator

- Mécanisme de détection de boucle
- Nouveau mode de génération. Complet et partiel.
- Le mode souhaité peut être sélectionné à partir de la boîte de dialogue de configuration.
- La génération complète ou partielle est désormais enregistrée avec le fichier de configuration du projet (xprj).
- Des types de données base manquant ont été ajoutés
- Réduction du temps de temps de génération.
- Nouvelle vérification avant la génération. Lorsque le fichier xmi est exporté sans stéréotype, la génération partielle échoue.
- Le projet peut être chargé à partir de la ligne de commande et généré automatiquement dans la même boucle.
- La ligne de commande est : `XiMulator x:\mypath\project.xprj`
- Le type de référence peut être ajouté à partir du diagramme de classes UML et généré.
- Ajout du niveau d'erreur dans le fichier journal.



# Nouveautés 2024

## Nouveautés dans le XiMulator

- Correction de l'erreur de génération lorsqu'un attribut avec valueRank 1 est créé dans un VariableType ou un ObjectType, sa définition de type est VariableArrayType au lieu de VariableType.
- Les fichiers NodeIdMap font désormais partie du fichier projet.
- Seuls les fichiers appartenant à un projet seront chargés
- Le dossier du projet est visible dans les paramètres de configuration
- On peut changer de dossier de projet.
- XiMulator est en Version 64 bits
- Le dossier du projet peut être visualisé et modifié dans les paramètres de configuration du projet



# Nouveautés 2024

## Correctif dans le Ximulator

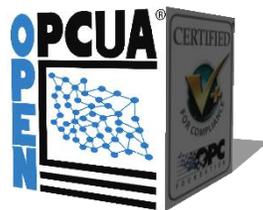
- Correction du problème de génération lié au mélange de langages de modélisation, le type de données de méthodes et/ou le type de données des attributs n'est pas correctement généré.
- Les méthodes avec des arguments de sortie ne sont pas générées correctement. Chaque argument de sortie est généré deux fois.
- Seul le fichier Nodeset spécifié dans le fichier projet est chargé
- Correction des problèmes dans l'utilisation de XiMulator en ligne de commande. Certains messages dans les boîtes de dialogue empêchaient une utilisation fluide.

# Questions ?



**Michel Condemine**

- OPC Foundation France (Technique)
- Directeur de **4CE Industry**
- Leader du projet OpenOpcUa
- MichelC@4CE-Industry.com



3<sup>ème</sup> édition prévue  
pour juin 2025