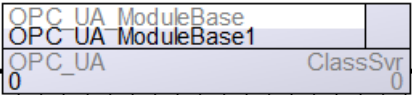


# OPC-UA\_ModuleBase

Diese Klasse dient als Basisklasse für OPC-UA Erweiterungs-Module der unterschiedlichsten Art. Die OPC-UA Diagnose-Klassen sind z.B. von dieser Klasse abgeleitet.



## Schnittstellen

### Clients

OPC-UA	Objektkanal zum OPC-UA-Server Objekt
--------	--------------------------------------

### Server

ClassSvr	Klassen-Server
----------	----------------

## Globale Methoden

<b>Init</b>	<p>Init-Methode: registriert dieses Modul im OPC_UA Server. Dabei wird durch die Übergabe von "multipleAllowed" eine wichtige Modul-Eigenschaft festgelegt - der Standardwert von "multipleAllowed" ist FALSE.  FALSE ... nur ein Modul dieser Art ist erlaubt  TRUE .... mehr als ein Modul dieser Art sind erlaubt</p> <p>Diese Methode überschreiben, wenn die Möglichkeit bestehen soll, mehr als ein Modul dieses Typs zu registrieren.</p>
<b>Background</b>	für zyklische Aufgaben
<b>InitModule</b>	<p>Kann zu Initialisierungszwecken überschrieben werden. Wird von OPC_UA während des Hochfahrens solange aufgerufen, bis #0 zurückgegeben wird.  Der OPC_UA Server startet erst, wenn von jedem Modul #0 zurückgegeben wurde! Somit kann diese Methode zum Verzögern des OPC_UA Server Startes verwendet werden.</p>
<b>InitProvider</b>	<p>Kann zu Provider-spezifischen Initialisierungszwecken überschrieben werden. Wird von OPC_UA während des Hochfahrens aufgerufen.</p>
<b>RegisterProvider</b>	<p>Bei Bedarf können hier OPC_UA-Provider registriert werden. Die Methode wird von OPC_UA während der Provider-Registrierungsphase aufgerufen.</p>
<b>AfterProviderInitialize</b>	Die Methode wird von OPC_UA aufgerufen, wenn alle Provider initialisiert wurden.
<b>OPC_UA_Thread</b>	<p>Diese Methode wird zyklisch indirekt vom OPC_UA-Thread aufgerufen. NUR überschreiben, wenn Funktionen synchron zum OPC_UA-Thread benötigt werden.  Andernfalls ist der Background-Task zu verwenden!</p>
<b>CB_activateDS</b>	<p>Callback Methode, muss bei Bedarf überschrieben werden. Wird aufgerufen, wenn ein OPC_UA Client ein Dataset an den Server übertragen und aktivieren möchte. Siehe auch Methoden-Doku in OPC_UA Klassen-Doku.</p>
<b>CB_preparedDS</b>	<p>Callback Methode, muss bei Bedarf überschrieben werden. Wird aufgerufen, wenn ein OPC_UA Client ein Dataset vom Server auslesen möchte. Siehe auch Methoden-Doku in OPC_UA Klassen-Doku.</p>
<b>CB_alarmList</b>	<p>Callback Methode, muss bei Bedarf überschrieben werden. Wird aufgerufen, wenn ein OPC_UA Client eine Liste aller anstehenden Alarmerfordert. Siehe auch Methoden-Doku in OPC_UA Klassen-Doku.</p>
<b>CB_fileSystem</b>	<p>Callback Methode, muss bei Bedarf überschrieben werden. Wird aufgerufen, wenn ein File durch OPC_UA geändert wurde. Siehe auch Methoden-Doku in OPC_UA Klassen-Doku.</p>
<b>CB_GetStringArray</b>	<p>Callback Methode, muss bei Bedarf überschrieben werden. Wird vom OPC_UA-Server aufgerufen, wenn ein String-Array auszulesen ist.</p> <p>IN: nodeld ... Pointer auf die Node-Id, für welche der String-Array ausgelesen werden soll (Struct vom Typ OPCUA_Nodeld). Die Node-Id kann für die Überprüfung via UA-Expert oder der Nodeset-XML eruiert werden.</p> <p>IN: list ... Pointer auf Pointer zum Retournieren der Liste mit den String-Pointern der einzelnen Strings im String-Array</p> <p>IN: listCount ... Pointer zum Retournieren der Anzahl von Strings im String-Array  OUT: retcode ... 0 = OK</p>

<b>CB_SetStringArray</b>	<p>Callback Methode, muss bei Bedarf überschrieben werden. Wird vom OPC-UA-Server aufgerufen, wenn ein String-Array zu schreiben ist.</p> <p>IN: nodeld ... Pointer auf die Node-Id, für welche der String-Array geschrieben werden soll (Struct vom Typ OPCUA_Nodeld). Die Node-Id kann für die Überprüfung via UA-Expert oder der Nodeset-XML eruiert werden.</p> <p>IN: list ... Pointer auf die Liste mit den String-Pointern der einzelnen Strings im String-Array</p> <p>IN: listCount ... Anzahl von Strings im String-Array</p> <p>OUT: retcode ... 0 = OK</p>
<b>OPC-UA_Run</b>	<p>Diese Methode wird zyklisch vom OPC-UA-Thread der OPC-UA-Klasse aufgerufen. Sie ruft wiederum die Methode "OPC-UA_Thread" auf, welche für eine modulspezifische Verwendung überschrieben werden kann.</p>