MTConnect - CMS/SCM

Consiste di vari Adapter (CUSTOM) ed un Agent (generico ed unico) che si configura come l'interfaccia esterna ai sistemi del cliente o di CMS/SCM (Cloudplugs)

MTC Agent

L'agent è comune a tutti i progetti ed è pressochè un prodotto standard dalle specifiche e dagli esempio predisposti dal consorzio MTC.

I file di configurazione sono contenuti nella cartella **..\AGENT\DATA\CONF**

In particolare l'agent ha bisogno di venire configurato tramite il file **Agent_ItemList.xml**

L'agent può anche autonomamente procedere alla conversione dei dati tra formato nativo e formato esposto (es da mm a metri, da INCH a mm...). Utile sia per correggere fattori di grandezza che per gestione macchine che abbianmo CNC in INCH e variabili PLC nativamente in MM: esempio

```
<DataItem category="SAMPLE" id="Axis_01_PosAct" type="POSITION" subType="ACTUAL"
nativeUnits="INCH" units="MILLIMETER" name="[mm] Quota attuale"/>
```

MTC Adapters PLC

L'adapter PLC è un programma unico che può comunicare con diversi PLC a seconda delle dll installate sulal macchina e del file di configurazione XML che verrà immesso nell'area **DATA\CONF**

E' importante verificare il file di configurazione applicativo **MTC_Adapter.exe.config*- poiché da alcuni parametri sarà impostabile il funzionamento produzione (adapter che resta in tray, all'utente non è permesso di ingrandirlo o chiuderlo, caricamento ed avvio automatico al lancio...) da quello di debug/testing con cui si possono verificare sia le varie configurazioni che eventuali problemi (avvio in modalità finestra, possibilità ingrandire/rimpicciolire, visualizzaizone finestra di dump messaggi inviati ad AGET, ...)

Ecco di seguito le chiavi principali da utilizzare **in produzione**

```
<add key="autoLoadConf" value="true"/>
<add key="autoStartOnLoad" value="true"/>
<add key="openDumpOnStart" value="false"/>
<add key="startMinimized" value="true"/>
<add key="windowCanMax" value="false"/>
<add key="trayClose" value="false"/>
```

Mentre queste sono quelle in versione **debug/diagnostica**

```
<add key="autoLoadConf" value="false"/>
<add key="autoStartOnLoad" value="true"/>
<add key="openDumpOnStart" value="true"/>
<add key="startMinimized" value="false"/>
<add key="windowCanMax" value="true"/>
<add key="trayClose" value="true"/>
```

L'adapter, a partire dalle versioni di settembre 2018, è istanziabile a scelta in modalità **MTC** (classica MTConnect) oppure **SOURS** (SCM Opc Ua Redis Server) destinato alla piattaforma cloud Maestro Connect. Per queste modalità prestare attenzione alla chiave "adpProto", che può appunto prendere i valori MTC o SOUR:

Ecco la chiave da usare per MTConnect

```
<add key="adpProto" value="MTC" />
```

Ecco la chiave da usare per OPC-UA

```
<add key="adpProto" value="SOURS" />
```

Altra modifica importante a partire da ottobre 2018 è la possibilità di impiegare il controllo dello status del sistema tramite le variabili della ACK_DW2 (vedere doc variabili

https://docs.google.com/spreadsheets/d/1SewqSJn941mKtzplY1ZtWe6W6Hw0Wtu_4RVjwIPo28w/edit#gid=848807258)

```
<add key="stChkEnabled" value="true"/>
```

In particolare per casistiche **MCT** viene verificato los tato dell'adapter, mentre per OPC-UA la "catena di invio" che c'è tra l'adapter ed il cloud SCM.

MTC-FANUC

- Primo adapter realizzato
- Utilizza le FOCAS 32 bit di fanuc (devono essere installate sul PC)

MTC-KVARA

- Adapter utilizzato da SCM
- Utilizza chiamate dirette tramite dll KVARA al PC/CN
- Riscritto per impiegare 4 macroaree dati (mappate) per interscambio flessibili di dati (byte/bit, Word, DWord, String)

MTC-OSAI

- impiega interfaccia SOAP di OSAI

MTC-SIEMENS

- gestione PLC Siemens
- va avviato da processo Synumeric (o da CMSControl)

MTC Adapters DB

Questi sono altri tipi di adapter realizzati per l'accesso a DB locali

MTC-ADB

- gestione db MySql custom per SavEnergy

Configurazione Agent

Va configurato un unico file di configurazione ovvero **Agent_ItemList.xml**

Agent_ItemList.xml

Vanno verificati i seguenti punti secondo la regola generale che quanto viene racchiuso tra i commentati `<!--` e `-->` sarà ignorato e NON presentato all'utente

- `Device` e `Description` iniziali: verificare e configurare correttamente
- assi: numero e tipo di ogni asse, in particolare distinguendo tra quelli di tipo `Linear` e quelli `Rotary` (consigliato copia e sostituzione indici da un asse già configurato)

- attenzione ai `Components`: non inserire più path di quanti gestiti dal CN (e configurati nell'adapter)
- sempre nei path commentare il vettore posizione punta utensile per i CN che non lo espongono (`Path_xx_PosAct[X,Y,Z,I,J,K]`)
- attenzione agli `Actuators`: non inserire più Unità Operatrici di quante gestite dal CN (e configurate nell'adapter)
- attenzione a `Sistems`: non inserire più oggetti (pompe, lubro, cooler) di quanti gestiti dal CN (e configurati nell'adapter)
- attenzione a `Sensoristica`: non inserire più variabili (tipicamente analogiche) (pompe, lubro, cooler) di quante gestite dal CN (e configurate nell'adapter)
- se si sono definiti oggetti ad esempio tramite variabili status, valori analogici, contatori nelle rispettive aree di memoria, è buona norma riorganizzarli come oggetti unici (che contengono tali variabili raggruppate)
- in data 2018.01.17 è stata aggiunta la possibilità (per OSAI) a livello di AGENT di ricevere la descrizione dell'asse (dinamica) e il processo di appartenenza (che sinora erano ignorate/cablate).

```
  <DataItem category="EVENT" id="Axis_01_Descr" type="MESSAGE" name="Denominazione
asse (dinamica)"/>
  <DataItem category="EVENT" id="Axis_01_MainProc" type="MESSAGE" name="[#]
Processo di appartenenza"/>
```

Configurazione ADAPTERS

I file da configurare in questo caso sono molteplici, e di seguito si indicano le note generali per la compilazione degli stessi

Adapter_ItemList.xml

Il file può essere impostato con l'adapter (qualsiasi), in modalità *debug/diagnostica* (vedere sopra)

1. Selezionare `SETUP > TEMPLATE CONF GENERATOR`
2. Caricare la configurazione attuale con `LOAD CONFIGURATION`
3. Si possono indicare quanti oggetti di ogni tipo si avranno nello specifico Adapter per la specifica macchina /CN
4. Ri-selezionare il tipo di adapter dal menù a tendina (se fosse già selezionato quello corretto cambiarlo e riselezionarlo, POTREBBE venire "persa" la selezione)
5. Salvare tramite il comando `SAVE CONFIGURATION`
6. Effetturare un ulteriore controllo e configurazione per gli assi (vanno specificati come `LINEAR / ROTARY` nella chiave `Axis_xx_Type`)

EsaKvara.ini

E' un file necessario solo per il controller ESA, ed è importante sia configurato per simulazione o produzione:

```
SysLink=SIMULATO ; in versione test/simulazione/sviluppo
;SysLink=NETWORK ; in versione reale/produzione
```

AlarmList.map

E' il file degli allarmi. Le righe commentate (ovvero che iniziano con #) saranno ignorate. Devono esserci tante righe quanti allarmi gestiti (tramite bit di stato ON/OFF) e ogni riga deve essere formattata secondo il seguente tracciato di esempio

```
000001|PLC|FAULT|[COD 001000] - 334 ERRORE COMUNICAZIONE PROFIBUS
000002|PLC|FAULT|[COD 001001] - 700 TABELLE IN MODIFICA
000003|PLC|FAULT|[COD 001002] - 404 BILANCIAMENTO ASSE Z INSUFFICIENTE
000004|PLC|FAULT|[COD 001003] - 403 SURRISCALDAMENTO ARMADIO ELETTRICO
000005|PLC|FAULT|[COD 001004] - 173 CAMBIARE LA BATTERIA
000006|PLC|FAULT|[COD 001005] - 407 UNO O PIU ASSI IN EXTRA CORSA
000007|PLC|FAULT|[COD 001006] - 406 PULSANTI DI EMERGENZA
```

Ovvero

- numero UNIVOCO incrementale 1..num tot allarmi
- sorgente PLC/CN (qui è PLC di default)
- tipo di allarme tra i due valori (FAULT/WARNING) che indicano guasto maggiore e segnalazione
- stringa di testo LIBERA che riporta il codice del problema e la sua descrizione (possibilmente come riportato a video all'operatore)

AnalogData.map

In questo file sono indicate le variabili analogiche che si vogliono rilevare. Il sistema è LIBERO (non ci sono vincoli di nomenclatura) e vengono restituiti dei SAMPLE con un nome nel formato `"AV_" + nome_variabile_indicata`

Il formato atteso è

```
001|Analog_01                    |NUM
002|Analog_02                    |NUM
003|Analog_03                    |NUM
```

Ovvero

- numero incrementale
- nome variabile (in output avrà previsso `"AV_"`)
- tipologia (NUM)

CounterList.map

In questo file sono indicate tutte le variabili di tipo contatore (da far gestire come ritentive all'adapter) che si vogliono rilevare. Il sistema **NON E' TOTALMENTE LIBERO*- (ci sono vincoli di nomenclatura) e vengono restituiti degli EVENTS con un nome identico all'originale

in particolare, nel seguente esempio TUTTE le variabili sono gestite "ad hoc" tranne i generici:

- Counter_xxx: è libero e rappresenta un contatore MONOTONO CRESCENTE, come un conta-km NON azzerabile (i valori pososno solo aumentare, se si "riporta indietro riprende ad aggiungere a partire dal valore ridotto)
- RTCounter_xxx: è libero e rappresenta un contatore REALTIME intero qualsiasi (permette di apssare anche dati che salgono e scendono, ad es un contapezzi secondario)

elenco di esempio:

```
001|ACC_TIME                    |HOURS
002|ACC_TIME_WORK               |HOURS
003|Path_01_PZ_TOT              |COUNT
004|Axis_01_DistDone            |METER
005|Axis_01_Invers              |COUNT
006|Axis_02_DistDone            |METER
007|Axis_02_Invers              |COUNT
008|Axis_03_DistDone            |METER
009|Axis_03_Invers              |COUNT
010|Axis_04_DistDone            |COUNT
011|Axis_04_Invers              |COUNT
012|Axis_05_DistDone            |COUNT
013|Axis_05_Invers              |COUNT
014|Axis_06_DistDone            |METER
015|Axis_06_Invers              |COUNT
016|UnOp_01_AccTime             |COUNT
017|VacPump_01_WrkTime          |HOURS
018|VacPump_02_WrkTime          |HOURS
019|VacAct_01_Count             |COUNT
020|VacAct_02_Count             |COUNT
021|Lubro_01_Num                |COUNT
022|SlittaTastatore_Count       |COUNT
023|SlittaMagazzino_01_Count    |COUNT
024|ProtMagazzino_01_Count      |COUNT
025|UnOp_01_NumCambiUT          |COUNT
026|Axis_01_AccTime             |COUNT
027|Axis_02_AccTime             |COUNT
028|Axis_03_AccTime             |COUNT
029|Axis_04_AccTime             |COUNT
030|Axis_05_AccTime             |COUNT
031|Axis_06_AccTime             |COUNT
032|Counter_001                 |COUNT
033|Counter_002                 |COUNT
034|Counter_003                 |COUNT
035|RTCounter_001               |COUNT
036|RTCounter_002               |COUNT
```

Ovvero

- numero incrementale
- nome variabile

- tipologia (HOURS/METER/COUNT, per i **Counter_xxx è SEMPRE COUNT**)

StatusList.map

In questo file sono indicate le variabili analogiche che si vogliono rilevare. Il sistema è LIBERO (non ci sono vincoli di nomenclatura) e vengono restituiti degli EVENTS con un nome nel formato `"ST_" + nome_variabile_indicata`, anche se ci sono "parole protette, tra cui stato protezioni (PROTECTION_STATUS), stato per Pompe, Cooler ed UnOp che verranno replicate nell'apposita variabile predisposta a standard per ogni item.

Il formato atteso è

```
001|PROTECTION_STATUS           |BIT
002|VacPump_01_Status           |BIT
003|VacPump_02_Status           |BIT
004|Cooler_01_Status            |BIT
005|Cooler_02_Status            |BIT
006|UnOp_01_Status              |BIT
007|Gen_Item_01                 |BIT
```

Ovvero

- numero incrementale
- nome variabile (in output avrà previsso `"ST_"`)
- tipologia (BIT)

IOT_Byte.map (ESAGV)

In questo file sono indicate le variabili gestite come BIT o BYTE nell'area IOT_Byte del CN

Il formato atteso è

```
0.0|IOT_EXEC                   |BOOL
0.1|IOT_HOLD                   |BOOL
0.2|IOT_EMG                    |BOOL
0.3|IOT_ALRM                   |BOOL
0.4|IOT_MACHO                  |BOOL
0.5|IOT_READY                  |BOOL
0.6|libero                     |BOOL
0.7|libero                     |BOOL
1.0|IOT_EXEC_A_01              |BOOL
1.1|IOT_EXEC_A_02              |BOOL
1.2|IOT_EXEC_A_03              |BOOL
1.3|IOT_EXEC_A_04              |BOOL
1.4|IOT_VAC_01                 |BOOL
1.5|IOT_VAC_02                 |BOOL
1.6|IOT_VAC_03                 |BOOL
1.7|IOT_VAC_04                 |BOOL
```

```
002|libero                    |BYTE
003|IOT_MODECN                |BYTE
004|IOT_OVRF                  |BYTE
005|IOT_OVRS                  |BYTE
006|IOT_LUB_01_STA            |BYTE
007|IOT_LUB_01_CNT            |BYTE
008|IOT_I_MD_01               |BYTE
009|IOT_I_MD_02               |BYTE
010|IOT_I_MD_03               |BYTE
011|IOT_I_MD_04               |BYTE
012|IOT_I_MD_05               |BYTE
013|IOT_I_MD_06               |BYTE
014|IOT_I_MD_07               |BYTE
015|IOT_I_MD_08               |BYTE
```

Ovvero

- numero incrementale nel formato `BYTE[.BIT]` (ovvero num byte + opzionalmente num BIT)
- nome variabile (passata in output UGUALE)
- tipologia (BOOL/BYTE)
- attenzione alla semantica: alcuni set informativi sono "blindati", ad esempio `IOT_I_MD_02` indica un contatore di `Load` del mandrino 2, `IOT_LUB_01_STA` è lo status dell'unità lubro 01, `IOT_LUB_01_CNT` è il conteggio impieghi dell'unità lubro 01, ...

IOT_WordList.map (ESAGV)

In questo file sono indicate le variabili gestite come Word (16bit) nell'area IOT_WordList del CN

Il formato atteso è

```
000|IOT_S_MD_01               |WORD
001|IOT_S_MD_02               |WORD
002|IOT_S_MD_03               |WORD
003|IOT_S_MD_04               |WORD
004|IOT_S_MD_05               |WORD
005|IOT_S_MD_06               |WORD
006|IOT_S_MD_07               |WORD
007|IOT_S_MD_08               |WORD
008|IOT_T_MD_01               |WORD
009|IOT_T_MD_02               |WORD
010|IOT_T_MD_03               |WORD
011|IOT_T_MD_04               |WORD
012|IOT_T_MD_05               |WORD
013|IOT_T_MD_06               |WORD
014|IOT_T_MD_07               |WORD
015|IOT_T_MD_08               |WORD
016|IOT_C_H_VAC_01            |WORD
017|IOT_C_H_VAC_02            |WORD
#018|IOT_C_H_VAC_03            |WORD
```

```
#019|IOT_C_H_VAC_04            |WORD
020|IOT_C_TC_01               |WORD
021|IOT_C_TC_02               |WORD
022|IOT_C_TC_03               |WORD
023|IOT_C_TC_04               |WORD
024|IOT_C_TC_05               |WORD
025|IOT_C_TC_06               |WORD
026|IOT_C_TC_07               |WORD
027|IOT_C_TC_08               |WORD
028|IOT_C_H_MD_01             |WORD
029|IOT_C_H_MD_02             |WORD
030|IOT_C_H_MD_03             |WORD
031|IOT_C_H_MD_04             |WORD
032|IOT_C_H_MD_05             |WORD
033|IOT_C_H_MD_06             |WORD
034|IOT_C_H_MD_07             |WORD
035|IOT_C_H_MD_08             |WORD
036|IOT_PGMR_A_01             |WORD
037|IOT_PGMR_A_02             |WORD
038|IOT_PGMR_A_03             |WORD
039|IOT_PGMR_A_04             |WORD
040|IOT_C_EXEC_A_01           |WORD
041|IOT_C_EXEC_A_02           |WORD
042|IOT_C_EXEC_A_03           |WORD
043|IOT_C_EXEC_A_04           |WORD
044|IOT_CN_MSG                |WORD
```

Ovvero

- numero incrementale nel formato `Word` (ovvero indice Word)
- nome variabile (passata in output UGUALE)
- tipologia (WORD)
- attenzione alla semantica: alcuni set informativi sono "blindati", ad esempio `IOT_S_MD_02` indica un la `Speed` del mandrino 2, `IOT_T_MD_01` è il numero del tool caricato sul mandrino 1, `IOT_PGMR_A_02` è il conteggio ripetizioni programma area 02, ...

IOT_DWordList.map (ESAGV)

In questo file sono indicate le variabili gestite come DoubleWord (32bit) nell'area IOT_DWordList del CN

Il formato atteso è

```
000|IOT_C_KU_AX_01            |DWORD
001|IOT_C_KU_AX_02            |DWORD
002|IOT_C_KU_AX_03            |DWORD
003|IOT_C_KU_AX_04            |DWORD
004|IOT_C_KU_AX_05            |DWORD
005|IOT_C_KU_AX_06            |DWORD
006|IOT_C_KU_AX_07            |DWORD
```

```
007|IOT_C_KU_AX_08              |DWORD
008|IOT_C_KU_AX_09              |DWORD
009|IOT_C_KU_AX_10              |DWORD
010|IOT_C_KU_AX_11              |DWORD
011|IOT_C_KU_AX_12              |DWORD
012|IOT_C_KU_AX_13              |DWORD
013|IOT_C_KU_AX_14              |DWORD
014|IOT_C_KU_AX_15              |DWORD
015|IOT_C_KINV_AX_01            |DWORD
016|IOT_C_KINV_AX_02            |DWORD
017|IOT_C_KINV_AX_03            |DWORD
018|IOT_C_KINV_AX_04            |DWORD
019|IOT_C_KINV_AX_05            |DWORD
020|IOT_C_KINV_AX_06            |DWORD
021|IOT_C_KINV_AX_07            |DWORD
022|IOT_C_KINV_AX_08            |DWORD
023|IOT_C_KINV_AX_09            |DWORD
024|IOT_C_KINV_AX_10            |DWORD
025|IOT_C_KINV_AX_11            |DWORD
026|IOT_C_KINV_AX_12            |DWORD
027|IOT_C_KINV_AX_13            |DWORD
028|IOT_C_KINV_AX_14            |DWORD
029|IOT_C_KINV_AX_15            |DWORD
030|IOT_POS_AX_01               |DWORD
031|IOT_POS_AX_02               |DWORD
032|IOT_POS_AX_03               |DWORD
033|IOT_POS_AX_04               |DWORD
034|IOT_POS_AX_05               |DWORD
035|IOT_POS_AX_06               |DWORD
036|IOT_POS_AX_07               |DWORD
037|IOT_POS_AX_08               |DWORD
038|IOT_POS_AX_09               |DWORD
039|IOT_POS_AX_10               |DWORD
040|IOT_POS_AX_11               |DWORD
041|IOT_POS_AX_12               |DWORD
042|IOT_POS_AX_13               |DWORD
043|IOT_POS_AX_14               |DWORD
044|IOT_POS_AX_15               |DWORD
045|IOT_C_KREV_MD_01            |DWORD
046|IOT_C_KREV_MD_02            |DWORD
047|IOT_C_KREV_MD_03            |DWORD
048|IOT_C_KREV_MD_04            |DWORD
049|IOT_C_KREV_MD_05            |DWORD
050|IOT_C_KREV_MD_06            |DWORD
051|IOT_C_KREV_MD_07            |DWORD
052|IOT_C_KREV_MD_08            |DWORD
053|IOT_PLC_MSG_00              |DWORD
054|IOT_PLC_MSG_01              |DWORD
055|IOT_PLC_MSG_02              |DWORD
056|IOT_PLC_MSG_03              |DWORD
057|IOT_PLC_MSG_04              |DWORD
```

```
058|IOT_PLC_MSG_05              |DWORD
059|IOT_PLC_MSG_06              |DWORD
060|IOT_PLC_MSG_07              |DWORD
061|IOT_PLC_MSG_08              |DWORD
062|IOT_PLC_MSG_09              |DWORD
063|IOT_PLC_MSG_10              |DWORD
064|IOT_PLC_MSG_11              |DWORD
065|IOT_PLC_MSG_12              |DWORD
066|IOT_FEED                    |DWORD
067|IOT_FEED_01                 |DWORD
068|IOT_FEED_02                 |DWORD
069|IOT_FEED_03                 |DWORD
```

Ovvero

- numero incrementale nel formato `DWord` (ovvero indice DWord)
- nome variabile (passata in output UGUALE)
- tipologia (DWORD)
- attenzione alla semantica: alcuni set informativi sono "blindati", ad esempio `IOT_C_KU_AX_04` indica un la `Distanza` totale percorsa dall'asse 4, `IOT_C_KINV_AX_01` è il numero (in K=migliaia) di inversioni dell'asse 1, `IOT_POS_AX_11` è la posizione dell'asse 11, ...

IOT_StringList.map (ESAGV)

In questo file sono indicate le variabili stringa gestite.

Il formato atteso è

```
001|IOT_PGMID_A1            |STRING
002|IOT_PGMID_A2            |STRING
003|IOT_PGMID_A3            |STRING
004|IOT_PGMID_A4            |STRING
```

Ovvero

- numero incrementale
- nome variabile
- tipologia (String)
- attenzione alla semantica: i set informativi sono "blindati" per il solo caso `IOT_PGMID_A1` che è il NOME completo del programma sull'area di memoria A...

FAQ e note

Alcune note finali su cosa verificare e alcune possibili cause di errori e problemi per misconfigurazioni

- Alarmlist.map:

    - è importante che NON siano duplicati i codici iniziali di ogni riga (devono essere univoci,s e fossero duplicati verrebbe preso SOLO il secondo record, ignorato il primo, con la conseguenza che il numero di allarmi sarebbero pari solo all'indice massimo, inferiore al numero di righe indicato)
    - è importante che ci siano un numero di allarmi pari ad un multiplo di 32 (eventualmente vuoti) per evitare problemi sulle dimensioni dei vettori di memoria nella copia tra dati ricevuto da PLC e area adapter. Mettere in ogni caso un banco di 32 bit di allarmi (vuoti) in aggiunta NON crea comunque problemi e potrebbe essere preso come misura precauzionale in caso non si popolasse il file con tutti i 1024 bit di allarme previsti
    - attenzione ad evitare "a capo" a fine file o caratteri strani che il sistema cerca di interpretarli e si blocca.

- AnalogData.map:

    - è importante ricordare che TUTTE le variabili analogiche saranno lette come INTERI 32bit e poit rasformate in valori decimali tramite un fattore di conversione definito nel file .config `<add key="fattDecVA" value="1000"/>` (quindi in questo caso i valori saranno divisi per un fattore 1000 e quindi con 3 decimali)
    - attenzione ad evitare "a capo" a fine file o caratteri strani che il sistema cerca di interpretarli e si blocca.

- CounterList.map:

    - è importante NON inserire contatori di tipo codificato con indici superiori agli oggetti definiti nell'Agent_ItemList.xml, ovvero ad esempio inserire `UnOp_02_NumCambiUT` se si fosse definita 1 sola unità operatrice (la variabile è il conteggio cambi utensili della 2° unità operatrice/mandrino)
    - attenzione ad evitare "a capo" a fine file o caratteri strani che il sistema cerca di interpretarli e si blocca.

- StatusList.map:

    - è importante NON inserire controlli di stato di tipo codificato con indici superiori agli oggetti definiti nell'`Agent_ItemList.xml`, ovvero ad esempio inserire `UnOp_02_Status` se si fosse definita 1 sola unità operatrice (la variabile è lo stato della 2° unità operatrice/mandrino) / si pososno inserire valori `EMPTY_xx` oppure commentare con un prefisso i valori da ignorare (ad es se non c'è 6° asse: `Axis_06_DistDone` --> `noAxis_06_DistDone`, `___Axis_06_DistDone`, `***Axis_06_DistDone`, ...)
    - attenzione ad evitare "a capo" a fine file o caratteri strani che il sistema cerca di interpretarli e si blocca.

- OSAI

    - attenzione che OSAI richeide che l'ordine degli assi impostati sia quello fornito dal controller SOAP, ovvero si deve guardare nell'adapter (in modalità debug) come sono espressi gli assi e manterene lo stesos ordine sul PLC, sull'adapter (che in ogni caso riordina rispetto a come invia i dati il webservice) e sull'agent
    - attenzione ad un altra peculiarità OSAI sempre per gli assi "dinamici": questo permette di inserire (nel file delle variabili dei contatori, **CounterList.map**) nomi delle VARIABILI (legate ai vari assi) in modo NON CONTIGUO/CONTINUO (es macchina 8646: contatore Distanza o numero inversioni

x asse 1..6 poi 7 e 11 SENZA gli assi 5,8,9,10). Inoltre per aderenza al formato di setup standard si tiene FISSA la prima parte dei contatori (i primi 6 assi, se poi non usati saranno filtrati e non gestiti a livello dell'agent). A livello di configurazione però, poiché il numero degli assi è slegato (per assi dinamici) dall'ID univoco dell'asse, è possibile avere ad esempio 8 contatori (la 8646 ha appunto 7 assi con ultimo asse che ha ID 11 e non definiti/usati "nel mezzo" gli assi 6,8,9,10).

- FANUC
    - modifica 2017.12.27: attenzione se ci sono machcine con numero di assi limitato (ad esempio con teste a 3assi, in cui non vanno le modalità 3D come TTip, teach, posizione...) perché alcune chiamate fanuc standard non funzionano (anzi proprio fanno crashare le focas) - per evitare problemi impostare la chiave **FanucLimit3D** a **true** (se manva o di default è **false**) che SALTA IN BLOCCO tali letture basate sulle **Focas1.cnc_rd5axmandt**; di conseguenza va commentato in XML dell'AGENT anche la parte corrispondente che NON verrà valorizzata --> **Path_[01..nn]_PosActX[XYZ|IJK]**