

Framework Promax.Industry40

2.0.0.18

Edited by Promax

1 Introduzione	1
2 Integrazione delle librerie	3
2.1 Collegamento al progetto host	3
2.2 Utilizzo nel codice	3
2.3 Avvio del server	4
2.4 Utilizzo di un device custom	5
2.5 Registrare i log del framework	6
2.6 Gestione degli ordini da interfaccia grafica	7
3 File di configurazione	9
3.1 Dati delle impostazioni	9
3.2 Dati delle variabili	12
3.3 Dati degli allarmi	13
3.4 Esempio di file	13
4 File dei dati degli eventi	15
5 Gestione dei database	17
5.1 Microsoft SQL Express	17
5.2 PostgreSQL	18
5.3 MySQL	20
5.4 Tabelle DB	20
5.4.1 Tabella Allarmi	21
5.4.2 Tabella Ordini/Produzione	21
5.4.3 Tabella Eventi	22
5.4.4 Tabella Stati	22
5.4.5 Tabella statistiche	22
5.4.6 Tabella Extra dati degli Ordini	22
5.4.7 Tabella Ricette/Programmi	22
6 Descrizione API Rest	25
6.1 Richieste principali	25
6.1.1 Richiesta state	26
6.1.2 Richiesta production	26
6.1.3 Richiesta orders	28
6.1.4 Richiesta order_extra_data	30
6.1.5 Richiesta events	31
6.1.6 Richiesta alarms	32
6.1.7 Richiesta connection	33
6.1.8 Richiesta statistics	33
6.1.9 Richiesta recipes	33
6.1.10 Richiesta settings	34

7 Storico delle versioni	35
7.1 Storico framework	35
7.2 Storico documentazione	36
Indice analitico	39

Capitolo 1

Introduzione

Il framework Promax.Industry40 è composto da un insieme di componenti che permette la connessione dei pannelli HMI su base PC ai sistemi gestionali o di supervisione dei macchinari per l'interconnessione richiesta per Industria 4.0. In questo documento verranno descritte le operazioni da effettuare per l'installazione del software, la configurazione del sistema ed il protocollo di comunicazione remota. Le caratteristiche salienti del software sono:

- Struttura modulare per facilitare future estensioni
- Utilizzo dei Database più comuni
- Semplicità d'uso e configurazione

Per il funzionamento del sistema le caratteristiche minime per l'inserimento in software esistenti sono:

- .NET Framework 4.7.2

Capitolo 2

Integrazione delle librerie

Il framework può essere integrato nel software host utilizzando le sole librerie senza avere il bisogno di altro. Questo permette di non avere bisogno di librerie di terze parti. Di seguito vengono descritti i passi per l'integrazione

2.1 Collegamento al progetto host

Per collegare le librerie al progetto host:

- Copiare la cartella delle librerie `Promax.Industry40` nella cartella del progetto del programma. Tutti i files sono nella cartella `..Salvataggi/Promax/LastVersions/PxIndustry40/Librerie` all'interno di questa cartella si trovano le cartelle delle due versioni:
 - **Light** - Versione senza la gestione dei database MySQL e PostgreSQL
 - **Full** - Versione Completa
- Collegare il riferimento alla libreria `'Promax.Industry40.dll'` al progetto di avvio dell'applicazione. ATTENZIONE!! Il progetto dovrà avere le seguenti impostazioni delle proprietà:
 - Target del framework minimo alla versione 4.7.2

2.2 Utilizzo nel codice

- Creare l'istanza di una classe di tipo **`Promax.Industry40.Industry40Manager`** o sua derivata comune a tutto il progetto e chiamare il metodo **`Initialize()`** dell'oggetto con i seguenti parametri:
 - Percorso assoluto dell'applicazione
 - Nome del file di configurazione per Industria 4.0
 - Flag per l'attivazione del log del server http
- Al termine dell'utilizzo dell'oggetto rilasciare le risorse con il metodo **`Uninitialize()`**

```

...
using Promax.Industry40;

namespace Test
{
    // Classe per l'uso del gestore
    public class I40Test
    {
        Industry40Manager mng = null;
        string appPath = "";

        public I40Test(string pth)
        {
            appPath = pth;
        }

        // Inizializzazione
        public void Initialize()
        {
            // Inizializza il gestore
            mng = new Industry40Manager();
            mng.Initialize(appPath, "test.cfg", false);
        }

        // Rilascio risorse
        public void Uninitialize()
        {
            mng.Uninitialize();
        }
    }
}
...

```

2.3 Avvio del server

Eseguite le operazioni di inizializzazione, è possibile avviare o arrestare il server di comunicazione remota utilizzando i metodi `Promax.Industry40.Industry40Manager.Start()` e `Promax.Industry40.Industry40Manager.Stop()`

```

...
using Promax.Industry40;

namespace Test
{
    // Classe per l'uso del gestore
    public class I40Test
    {
        Industry40Manager mng = null;
        string appPath = "";

        // Costruttore
        public I40Test(string pth)
        {
            appPath = pth;
        }

        // Inizializzazione
        public void Initialize()
        {
            // Inizializza il gestore
            mng = new Industry40Manager();
            mng.Initialize(appPath, "test.cfg", false);
        }

        // Rilascio risorse
        public void Uninitialize()
        {
            mng.Uninitialize();
        }

        // Avvio del server
        public void Start()
        {
            mng.Start();
        }

        // Arresto del server
        public void Stop()
        {
            mng.Stop();
        }
    }
}
...

```

```

    }
}
...

```

In questo modo dai software remoti sarà possibile accedere ai dati messi a disposizione del sistema. I dati andranno inseriti utilizzando i metodi messi a disposizione della classe Promax.Industry40.Industry40Manager

2.4 Utilizzo di un device custom

Nel caso si volesse impostare un gestore per la comunicazione diverso da quello definito nel file di configurazione, si potrà utilizzare il metodo **SetDevice** passando un oggetto derivato dall'interfaccia **IDevice**, il quale verrà richiamato al posto di quello impostato di default

```

...
using Promax.Industry40;

namespace Test
{
    // Classe per la comunicazione con un device
    public class TestDev : IDevice
    {
        ...
    }

    // Classe per l'uso del gestore
    public class I40Test
    {
        Industry40Manager mng = null;
        string appPath = "";

        // Costruttore
        public I40Test(string pth)
        {
            appPath = pth;
        }

        // Inizializzazione
        public void Initialize()
        {
            // Inizializza il gestore
            mng = new Industry40Manager();
            mng.Initialize(appPath, "test.cfg", false);

            // Assegna il device al gestore
            TestDev dev = new TestDev();
            mng.SetDevice(dev);
        }

        // Rilascio risorse
        public void Uninitialize()
        {
            mng.Uninitialize();
        }

        // Avvio del server
        public void Start()
        {
            mng.Start();
        }

        // Arresto del server
        public void Stop()
        {
            mng.Stop();
        }

        // Metodo da richiamare ciclicamente
        // per la lettura dei dati dal device
        public void ReadDataFromDevice()
        {
            mng.ReadData();
        }
    }
}
...

```

A questo punto il gestore è pronto per essere utilizzato. Per impostare i dati degli eventi o degli stati, sono a disposizione questi due tipi di gestione:

- Chiamare ciclicamente il metodo 'ReadData' dell'oggetto (Vedi codice precedente). In questo modo il gestore richiederà al device impostato dal file di configurazione o manualmente, i valori delle variabili presenti nel file di configurazione
- Aggiungere manualmente i dati degli stati, degli eventi e delle impostazioni degli ordini e produzioni tramite i metodi dell'oggetto

2.5 Registrare i log del framework

Per registrare i log del framework, assegnare alla classe statica **Px40LoggerService** un oggetto derivato dall'interfaccia **IPx40Logger**, in modo da poter registrare tutti gli eventi e tenerne traccia per il debug del sistema

```
using Promax.Industry40;

namespace Test
{
    // Classe per la comunicazione con un device
    public class TestDev : IDevice
    {
        ...
    }

    // Classe per il log della libreria
    public class TestLogger : IPx40Logger
    {
        ...
    }

    // Classe per l'uso del gestore
    public class I40Test
    {
        Industry40Manager mng = null;
        string appPath = "";

        // Costruttore
        public I40Test(string pth)
        {
            appPath = pth;
        }

        // Inizializzazione
        public void Initialize()
        {
            // Inizializza il gestore
            mng = new Industry40Manager();
            mng.Initialize(appPath, "test.cfg", false);

            // Assegna il device al gestore
            TestDev dev = new TestDev();
            mng.SetDevice(dev);

            // Assegna il logger al gestore
            TestLogger log = new TestLogger();
            Px40LoggerService.Logger = log;
        }

        // Rilascio risorse
        public void Uninitialize()
        {
            mng.Uninitialize();
        }

        // Avvio del server
        public void Start()
        {
            mng.Start();
        }

        // Arresto del server
        public void Stop()
        {
            mng.Stop();
        }
    }
}
```

```

    }

    // Metodo da richiamare ciclicamente
    // per la lettura dei dati dal device
    public void ReadDataFromDevice()
    {
        mng.ReadData();
    }
}
...

```

IMPORTANTE: Il salvataggio dei tipi di log può essere filtrato con un'impostazione nel file di configurazione

2.6 Gestione degli ordini da interfaccia grafica

Per la gestione dei dati di produzione sono presenti dei controlli i tipi di interfaccia Forms e WPF. Questi controlli sono implementati rispettivamente nei namespaces **Promax.Industry40.Ui.Forms** e **Promax.Industry40.WPF**. Per poter utilizzare questi controlli all'interno delle finestra del programma dovranno:

- Aver impostata la proprietà **I40Manager** con l'oggetto **Industry40Manager** o derivato creato in precedenza
- Aver collegato l'evento **NotifyManageData** per la comunicazione delle operazioni. Questo evento viene inviato con un oggetto di tipo **PxIndustry40EventArgs** che ha le seguenti proprietà:
 - Cancel – Valore booleano che viene controllato al rientro per abortire l'operazione
 - Action – Identificativo dell'operazione da eseguire. È di tipo **OrderAction** e può assumere i seguenti valori:

Identificativo	Valore	Descrizione
None	0	Nessun comando
SelectionChanged	4	Comando per la segnalazione del cambio di selezione nella lista ordini
Select	5	Comando per la selezione di un programma dell'ordine. In questo caso la proprietà Order dovrà essere valida. Per annullare l'operazione impostare a True la proprietà Cancel
ProductedExceptionSet	6	Comando per impostare i dati delle quantità da produrre e già prodotte. Anche in questo caso la proprietà Order dovrà essere valida per poter prelevare i dati delle quantità prodotta e da produrre
ProductedExceptionRequest	7	Comando per la richiesta della quantità prodotta al momento. Viene utilizzata per quando gli ordini vengono sospesi o terminati
RequestTranslation	8	Comando per la traduzione di alcune stringhe
RequestYesNoOperation	9	Comando per la richiesta di risposta di conferma di un'operazione da eseguire

- ExchangeData – Oggetto per lo scambio dei dati. Deve essere reimpostato per alcune operazioni
- Order – Dati dell'ordine se da impostare

Capitolo 3

File di configurazione

Il file di configurazione viene utilizzato dal framework per il funzionamento del sistema con le impostazioni definite dall'utente. E' in formato XML, e può essere generato in due modi:

- Tramite un editor di testo
- Utilizzando il software PxIndustry40Configurator.

Il file deve avere la seguente struttura:

- **Data** – Nodo principale
 - **Settings** – Nodo principale per i dati delle impostazioni
 - * **Setting** – Nodo per il singolo dato delle impostazioni. Questo nodo deve avere i seguenti attributi:
 - **Name** – Nome dell'impostazione. (Per i nomi disponibili vedere il paragrafo 'Dati delle impostazioni')
 - **Value** – Valore dell'impostazione. (Per i valori vedere il paragrafo 'Dati delle impostazioni')
 - **Variables** – Nodo principale per i dati delle variabili
 - * **Variable** – Nodo per il singolo dato di una variabile. Questo nodo deve avere gli attributi descritti al 'Dati delle variabili'
 - **Alarms** – Nodo principale per i dati degli allarmi
 - * **Alarm** – Nodo per il singolo dato di un allarme. Questo nodo deve avere gli attributi descritti al 'Dati degli allarmi'

3.1 Dati delle impostazioni

In questo paragrafo vengono descritti i dati delle impostazioni disponibili al momento. La tabella dei dati è la seguente:

Nome	Opt	Descrizione	Default
Machineld		Id della macchina in formato stringa	
UseCrypt	si	Flag per l'uso di dati utente e password criptati True se i dati sono criptati, altrimenti False. Il valore di default è False	

Nome	Opt	Descrizione	Default
ServerType		Tipo di server da attivare Al momento i valori possono essere <ul style="list-style-type: none"> • NONE • HTTP 	
ListenPort		Porta di comunicazione per i server. Va inserito il valore per la comunicazione TCP/IP del server Tipicamente i valori sono: <ul style="list-style-type: none"> • HTTP: 8010 	
ServerUser		Nome dell'utente per l'accesso al server. Se Utente e password sono valori vuoti, l'accesso è libero	
ServerPassword		Password dell'utente per l'accesso al server	
PageLanguage	si	Linguaggio delle pagine del server (solo per server http). Identifica la cartella per le pagine HTML se si accede con un browser Il valore predefinito è it	
FlagsEnableLog	si	Flags per l'abilitazione dei tipi di log I valori validi per i vari tipi di log sono: <ul style="list-style-type: none"> • 0x0001 – INFO • 0x0002 – WARNING • 0x0004 – ERROR • 0x0008 - FATAL 	0
EnableLogServer	si	Abilita la gestione del log del server Il valore predefinito è True	
DBManagerType		Tipo di manager per il Database Al momento i valori possono essere: <ul style="list-style-type: none"> • XML • POSTGRESQL • MSSQL • MYSQL • SQLITE • EXCEL 	
DBDatabaseName		Nome del database nel server DB oppure nome del file per i tipi XML, SQLITE ed EXCEL	
DBServerName		Nome del server remoto sul quale è installato o l'indirizzo IP del server remoto	
DBPort		Porta di comunicazione TCP/IP del server DB. È valido solo per i server e non per il gestore XML. Le porte di default per la comunicazione per i vari DB sono: <ul style="list-style-type: none"> • PostgreSQL : 5432 • MsSQL : 1433 • MySql : 3306 	

Nome	Opt	Descrizione	Default
DBUser		Nome dell'utente per l'accesso al DB Il valore deve essere presente anche tra gli utenti del DB	
DBPsw		Password dell'utente per l'accesso al DB e per il tipo EXCEL Il valore deve essere associato ad un utente nel DB	
DeviceType		Tipo di device collegato per la lettura dei dati delle variabili e degli eventi. Al momento i valori possono essere: <ul style="list-style-type: none"> • VIRTUAL • PROMAX • PROMAX_SYS 	
DeviceAddress		Indirizzo TCP/IP per la comunicazione con il device	
DevicePort		Porta di comunicazione TCP/IP per la comunicazione con il device Per i devices Promax il valore di default è 6003	
DeviceTimeout		Valore di timeout per la comunicazione con il device. Il valore deve essere in ms	
RequestInterval	si	Tempo di intervallo tra le richieste al device Il valore predefinito è 2000	
ReadAllVariablesSameTime	si	Flag per la lettura contemporanea di tutte le variabili. Se impostata a 0, bisognerà chiamare la funzione di aggiornamento ciclicamente Il valore predefinito è 0	
SendDataToDeviceAtStart	si	Deve essere impostato a True se all'avvio alcuni dati devono essere inviati al device	PROMAX_SYS
MonitorFile		Nome del file con i dati di monitor generati dalla compilazione del progetto VTB2. ATTENZIONE!! Viene utilizzato solo dal device PROMAX_SYS	PROMAX_SYS
EventMessagesFile		Nome del file per i dati delle descrizioni degli eventi	Tutti
RecipesPathIsAbsolute		Flag per il percorso assoluto delle ricette. Se impostato a True, il valore di 'RecipesPath' deve essere un percorso assoluto, altrimenti sarà relativo al percorso dell'applicazione	False
RecipesPath	si	Percorso relativo alla cartella base per la ricerca dei file delle ricette	Tutti
RecipesFileExtension	si	Estensione del file delle ricette per la restituzione della lista	Tutti
SerializerType	si	Tipo di serializzazione per la restituzione dei dati richiesti. Al momento i valori possono essere: <ul style="list-style-type: none"> • XML • JSON 	Tutti
BackupType		Tipo di gestore per il backup Al momento i valori possono essere: <ul style="list-style-type: none"> • LOCAL 	
BackupSource		Percorso sorgente per il backup	
BackupCustomerName		Nome del cliente	
BackupMachineName		Nome del macchinario	
BackupMachineSn		Numero seriale del macchinario	
BackupServerAddr		Indirizzo del server per il backup	

Nome	Opt	Descrizione	Default
BackupServerPort		Porta IP del server per il backup	
BackupServerUser		Nome utente per l'accesso al server del backup	
BackupServerPsw		Password per l'accesso al server del backup	
BackupServerDest		Cartella di destinazione del backup	
ViewCustomColumn	si	Visualizza la colonna custom nei controlli integrati nella libreria Deve essere il nome di un campo nei dati custom. Vedere il capitolo 7.1 comando procuttion add per chiarimenti	
QuantityResolution	si	Risoluzione delle quantità per la conversione sui macchinari Se la macchina lavora in mm ed i dati sono in mt, impostare 1000	1
ViewSortingCriterion	si	Definisce il criterio di ordinamento della lista degli ordini nella visualizzazione nelle finestre della libreria I criteri al momento utilizzabili sono: <ul style="list-style-type: none"> • ID – Id dell'ordine • STATE – Stato dell'ordine 	
ViewHideColumns	si	Definizione delle colonne da nascondere nella visualizzazione nelle finestre della libreria. La stringa deve essere composta dai nomi delle colonne separati dal carattere I nomi delle colonne utilizzabili sono: <ul style="list-style-type: none"> • colArticle • colX • colY • colTodo • colDone 	

3.2 Dati delle variabili

Gli attributi disponibili per le variabili sono:

Nome	Descrizione	Default
Type	Tipo di valore. Al momento viene impostato solo Var	Var
Name	Nome della variabile che verrà visualizzata come stato da inserire nel database	
TargetName	Nome della variabile target del device letta dal file Monitor. (Utilizza il file di monitor per le apparecchiature Promax)	
BaseAddress	Indirizzo di base per le variabili puntatore ATTENZIONE!! : Per utilizzare i dati nel modo corretto bisognerà: <ul style="list-style-type: none"> • Definire una variabile puntatore con un nome di riferimento(Es: V_DATA_PTR) e l'attributo BaseAddress impostato • Definire le variabili che fanno riferimento al puntatore con il valore di TargetName uguale al nome della variabile puntatore e l'indice del valore LONG della variabile puntatore come valore di Index 	Opz
OffsetFromBaseAddress	Offset da aggiungere all'indirizzo base per la lettura dei dati dalla memoria	0

Nome	Descrizione	Default
Index	Indice del valore delle variabili associate ad una variabile puntatore	-1
IsVirtual	Flag per la definizione di una variabile virtuale, che non verrà letta dal controllo ma il suo valore potrà essere impostato direttamente da software	false
Resolution	Risoluzione del valore per la conversione	1
ShowState	Flag per la visualizzazione del dato negli stati	true
StaticName	Nome del dato della statistica per il calcolo dei tempi macchina. Questi nomi vanno messi per identificare i valori da utilizzare per il calcolo dei tempi	

Nella tabella seguente vengono descritti i nomi predefiniti delle variabili. I valori sono:

Nome	Descrizione
RunState	Nome della variabile per lo stato di run
StandbyState	Nome della variabile per lo stato di standby
AlarmsState	Nome della variabile per lo stato di arresto
PiecesOk	Nome della variabile per il numero di pezzi eseguiti
PiecesReject	Nome della variabile per il numero di pezzi scartati

Per le variabili utilizzate per le statistiche (attributo **StatisticName**), i valori predefiniti sono:

Nome	Descrizione
work_time	Variabile per il conteggio del tempo di lavoro
stop_time	Variabile per il conteggio del tempo di arresto

3.3 Dati degli allarmi

Gli attributi disponibili per i dati degli allarmi sono:

Nome	Descrizione
Id	Identificativo dell'allarme
Variable	Variabile dalla quale leggere lo stato del bit associato all'allarme. La variabile deve essere definita nella lista Variables
Bit	Posizione del bit per l'allarme nella variabile (Partendo da 0)
Text	Testo dell'allarme

3.4 Esempio di file

Di seguito un esempio di file di configurazione:

```
<?xml version="1.0" encoding="utf-8"?>
<Data>
  <Settings>
    <Setting Name="ServerType" Value="HTTP"/>
    <Setting Name="ListenPort" Value="8010"/>
    <Setting Name="PageLanguage" Value="it"/>
    <Setting Name="ServerUser" Value="test"/>
    <Setting Name="ServerPsw" Value="test"/>
    <Setting Name="DBManagerType" Value="POSTGRESQL"/>
    <Setting Name="DBDatabaseName" Value="system"/>
  </Settings>
</Data>
```

```
<Setting Name="DBServerName" Value="127.0.0.1"/>
<Setting Name="DBPort" Value="5432"/>
<Setting Name="DBUser" Value="test"/>
<Setting Name="DBPsw" Value="test"/>
<Setting Name="DeviceType" Value="PROMAX"/>
<Setting Name="DeviceAddress" Value="10.0.0.80"/>
<Setting Name="DevicePort" Value="6003"/>
<Setting Name="DeviceTimeout" Value="1000"/>
<Setting Name="ViewSortingCriterion" Value="state"/>
<Setting Name="ViewHideColumns" Value="colX|colY"/>
</Settings>
<Variables>
  <Variable Type="Var" Name="V_VERSION" Index="0" TargetName="SYS_VER" ShowState="false"/>
  <Variable Type="Var" Name="V_RUN" Index="0"/>
  <Variable Type="Var" Name="V_TOTAL_TIME" Index="0"/>
  <Variable Type="Var" Name="V_TOTAL_METERS" Index="0" Resolution="1000"/>
  <Variable Type="Var" Name="V_WORK_TIME" Index="0" StatisticName="CURRENT_WORKTIME"/>
  <Variable Type="Var" Name="V_STOP_TIME" Index="0" StatisticName="CURRENT_STOPTIME"/>
  <Variable Type="Var" Name="V_DATA_PTR" BaseAddress="5467700" OffsetFromBaseAddress="4"/>
  <Variable Type="Var" Name="V_DATA_01" TargetName=V_DATA_PTR" Index="0"/>
  <Variable Type="Var" Name="V_DATA_02" TargetName=V_DATA_PTR" Index="1"/>
  <Variable Type="Var" Name="V_DATA_03" TargetName=V_DATA_PTR" Index="2"/>
  <Variable Type="Var" Name="V_ALARMS" Index="0"/>
  <Variable Type="Var" Name="V_ALARMS_2" IsVirtual="true" Index="0"/>
</Variables>
<Alarms>
  <Alarm Id="1" Variable="V_ALARMS" Bit="0" Text="AL_CFG_UPDOWN_DX"/>
  <Alarm Id="2" Variable="V_ALARMS" Bit="1" Text="AL_CFG_UPDOWN_SX"/>
  <Alarm Id="3" Variable="V_ALARMS" Bit="2" Text="AL_EMCY_UPDOWN_DX"/>
  <Alarm Id="4" Variable="V_ALARMS" Bit="3" Text="AL_EMCY_UPDOWN_SX"/>
  <Alarm Id="5" Variable="V_ALARMS" Bit="4" Text="AL_UPDOWN_DX_DISABLE"/>
  <Alarm Id="6" Variable="V_ALARMS" Bit="5" Text="AL_UPDOWN_SX_DISABLE"/>
  <Alarm Id="7" Variable="V_ALARMS_2" Bit="0" Text="AL_EMERGENCY"/>
  <Alarm Id="8" Variable="V_ALARMS_2" Bit="1" Text="AL_SYS_ERROR"/>
</Alarms>
</Data>
```

Capitolo 4

File dei dati degli eventi

I files dei messaggi da associare ai codici degli eventi sono in formato XML, e possono essere generati tramite un editor di testo Il file deve avere la seguente struttura:

- EventsMessages – Nodo principale
 - EventMessage – Nodo principale per i dati dei codici e messaggio. Questo nodo deve avere i seguenti attributi:
 - * Id – Id dell'evento
 - * SubId – Sottotipo di evento da associare all'Id come delle categorie
 - * Message – Messaggio associato all'evento

Le descrizioni degli eventi saranno associate automaticamente dal sistema quando gli eventi saranno inseriti nel database. Di seguito un esempio di file XML:

```
<?xml version="1.0" encoding="utf-8"?>
<EventsMessages>
  <EventMessage Id="1" SubId="0" Message="Avvio programma da pulsante" />
  <EventMessage Id="2" SubId="0" Message="Arresto programma da pulsante" />
  <EventMessage Id="2" SubId="1" Message="Arresto per allarme motori" />
  <EventMessage Id="2" SubId="2" Message="Arresto per emergenza" />
  <EventMessage Id="100" SubId="0" Message="Avvio sistema" />
</EventsMessages>
```


Capitolo 5

Gestione dei database

Per l'installazione dei gestori database da poter utilizzare con questo sistema, verranno descritte nel dettaglio le procedure di installazione per ognuno. I gestori che si possono utilizzare nella libreria light sono:

- XML (Predefinito)
- Excel
- SQLite
- Microsoft SQL Express

Nella libreria estesa e nel software PxIndustry40Manager sono inoltre compresi:

- PostgreSQL
- MySql

Per il gestore XML non è prevista installazione perché è un normale formato di file, mentre per SQLite, la gestione è affidata ad una DLL che normalmente è già integrata nel sistema operativo. Per il gestore Excel, non è necessario avere Office installato.

5.1 Microsoft SQL Express

Per l'installazione del server DB Microsoft SQL Express sulla macchina locale, bisogna seguire i seguenti passi:

- Installare il server dal file di setup
- Creare il database per il salvataggio dei dati ed inserire il nome nel file di configurazione del server

Per l'attivazione della connessione da remoto, bisogna abilitare il server ed il firewall per la comunicazione da remoto. Per fare questo, i passi da seguire sono:ù

- Avviare SQL Server configuration manager, normalmente installato nella cartella C:\Windows\SysWOW64. Il file si chiama SQLServerManager[XX].msc

- Selezionare ed espandere la voce Configurazione di rete SQLServer nella vista ad albero sulla sinistra della finestra
- Selezionare l'elemento Protocolli per SQLEXPRESS
- Abilitare il protocollo TCP/IP utilizzando il menu di contesto (Tasto destro del mouse)
- Aprire la finestra delle proprietà del Protocollo TCP/IP utilizzando il menu di contesto (Come fatto precedentemente)
- Selezionare il tab Indirizzi IP
- Scorrere la lista fino in fondo alla voce **IPALL**
- Impostare il valore Porta TCP/IP al valore desiderato (Normalmente 1433)
- Premere il pulsante Applica
- Chiudere SQL server configuration manager
- Aprire la porta TCP per le connessioni da remoto per MsSQLEXPRESS tramite Windows Defender Firewall aggiungendo una nuova regola alle connessioni in ingresso. Per fare questo, seguire i seguenti passi:
 - Avviare il configuratore di Windows Defender Firewall da Pannello di controllo->Sistema e sicurezza->Windows Defender Firewall
 - Selezionare Impostazioni avanzate nella lista dei comandi sulla sinistra della finestra
 - Selezionare il 'Regole connessioni in entrata' nella lista delle regole nella parte sinistra della finestra
 - Eseguire il comando Azione->Nuova Regola dal menu della finestra
 - Nel dialogo che si aprirà per la creazione guidata delle regole selezionare Porta per il tipo di regola e premere Avanti
 - Nel pannello successivo selezionate TCP come protocollo di comunicazione e selezionare Porte locali specifiche. Inserire la porta di comunicazione del Server SQLEXPRESS. Il numero della porta deve essere quello impostato in precedenza
 - Premere il tasto Avanti fino all'ultima pagina
 - Nell'ultima pagina inserire il nome della regola in modo da poterla trovare con più facilità
 - Premere Fine per il salvataggio delle impostazioni
- Fermare e poi fare ripartire il servizio SQLEXPRESS dal gestore servizi nel Pannello di controllo->Sistema e sicurezza-> Strumenti di amministrazione

5.2 PostgreSQL

Per l'installazione del server DB PostgreSQL sulla macchina locale, è necessario seguire i seguenti passi:

- Installare il server dal file di setup
- Alla richiesta dei dati amministratore "postgres" inserire:
 - **Password** – La password per l'amministratore
- Avviare il programma di gestione DB pgAdmin dal menu di Windows
- Accedere al server utilizzando i dati di User e Password per l'amministratore inseriti in fase di setup (User: postgres - Password: [quella inserita in fase di setup])
- Creare un utente in Login/Group Roles per l'accesso al DB della macchina con le seguenti proprietà:
 - Tabella **General** – Inserire il nome utente

- Tabella **Definitions** – Inserire la password
- Tabella **Privileges** – Abilitare le seguenti opzioni:
 - * Can login
 - * Create database
 - * Inherit rights from the parent roles (Già abilitato di default)
- Al termine delle impostazioni salvare premendo il tasto **Save**
- Creare il database per la gestione dei dati in Databases con le seguenti proprietà:
 - Tabella **General** – Inserire il nome del database e in **Owner** l'utente creato in precedenza
 - Tabella **Definition** – Selezionare le seguenti voci:
 - * **Tablespace** – Selezionare pg_default
 - Al termine delle impostazioni salvare premendo il tasto **Save**

Ora è possibile utilizzare il database sulla macchina in locale

Per l'attivazione della connessione da remoto, bisogna abilitare il server ed il firewall per la comunicazione da remoto. Per fare questo, i passi da seguire sono:

- Modificare il file di configurazione di PostgreSQL. Per fare questo seguire i seguenti passi:
 - Andare nella cartella Data dentro la cartella di installazione di PostgreSQL
 - Aprire il file pg_hba.conf con un editor di testo
 - Nel file aggiungere questa stringa:

```
host all all <range_ip_addr> trust
```

dove **range_ip_addr** è il range di indirizzi abilitati per la connessione. La stringa può essere:
 - * all – In questo modo sono abilitati tutti gli indirizzi 10.0.0.1/32 – Così vengono abilitati tutti gli indirizzi da 10.0.0.1 a 10.0.0.255. Il valore 32 è il numero dei bit di abilitazione
- Salvare il file

Aprire la porta TCP per le connessioni da remoto per PostgreSQL tramite Windows Defender Firewall aggiungendo una nuova regola alle connessioni in ingresso. Per fare questo, seguire i seguenti passi:

- Avviare il configuratore di Windows Defender Firewall da Pannello di controllo->Sistema e sicurezza->Windows Defender Firewall
- Selezionare Impostazioni avanzate nella lista dei comandi sulla sinistra della finestra
- Selezionare il **Regole connessioni in entrata** nella lista delle regole nella parte sinistra della finestra
- Eseguire il comando Azione->Nuova Regola dal menu della finestra
- Nel dialogo che si aprirà per la creazione guidata delle regole selezionare Porta per il tipo di regola e premere Avanti
- Nel pannello successivo selezionate TCP come protocollo di comunicazione e selezionare Porte locali specifiche. Inserire la porta di comunicazione del Server PostgreSQL. Il numero della porta deve essere quello del server impostata durante la configurazione del server DB (Normalmente 5432)
- Premere il tasto Avanti fino all'ultima pagina
- Nell'ultima pagina inserire il nome della regola in modo da poterla trovare con più facilità
- Premere Fine per il salvataggio delle impostazioni
- Fermare e poi fare ripartire il servizio PostgreSQL dal gestore servizi nel Pannello di controllo->Sistema e sicurezza-> Strumenti di amministrazione

5.3 MySQL

Per l'installazione del server DB MySQL sulla macchina locale, bisogna seguire i seguenti passi:

- Installare il server dal file di setup
- Creare un utente per l'accesso al DB
- Creare il database per il salvataggio dei dati ed inserire il nome nel file di configurazione del server

Per l'attivazione della connessione da remoto, bisogna abilitare il server ed il firewall per la comunicazione da remoto. Per fare questo, i passi da seguire sono:

- Avviare il software MySQL Workbench
- Configurare una connessione per la ricezione dei dati da remoto
- Aprire la porta TCP per le connessioni da remoto per MySQL tramite Windows Defender Firewall aggiungendo una nuova regola alle connessioni in ingresso. Per fare questo, seguire i seguenti passi:
 - Avviare il configuratore di Windows Defender Firewall da Pannello di controllo->Sistema e sicurezza->Windows Defender Firewall
 - Selezionare Impostazioni avanzate nella lista dei comandi sulla sinistra della finestra
 - Selezionare il **Regole connessioni in entrata** nella lista delle regole nella parte sinistra della finestra
 - Eseguire il comando Azione->Nuova Regola dal menu della finestra
 - Nel dialogo che si aprirà per la creazione guidata delle regole selezionare Porta per il tipo di regola e premere Avanti
 - Nel pannello successivo selezionate TCP come protocollo di comunicazione e selezionare Porte locali specifiche. Inserire la porta di comunicazione del Server MySQL. Il numero della porta deve essere quello impostato in precedenza
 - Premere il tasto Avanti fino all'ultima pagina
 - Nell'ultima pagina inserire il nome della regola in modo da poterla trovare con più facilità
 - Premere Fine per il salvataggio delle impostazioni
- Fermare e poi fare ripartire il servizio MySQL dal gestore servizi nel Pannello di controllo->Sistema e sicurezza-> Strumenti di amministrazione

5.4 Tabelle DB

Le tabelle da gestire nel database collegato saranno le seguenti:

- Allarmi
- Ordini e produzioni
- Eventi
- Stati
- Statistiche
- Ricette/Programmi

Alcuni dati possono essere differenti secondo il gestore del database utilizzato. La prossima tabella descrive l'associazione dei dati secondo i vari gestori:

Tipo dato	MsSQL	PostgreSQL	MySQL	SQLite
KEY	INT NOT NULL PRIMARY KEY IDENTITY(1,1)	SERIAL PRIMARY KEY	SERIAL PRIMARY KEY	INTEGER PRIMARY KEY AUTOINCREMENT
BOOLEAN	BIT	INT	BOOLEAN	INT
DATE	DATETIME	timestamp without time zone	TIMESTAMP	DATETIME

5.4.1 Tabella Allarmi

La tabella degli allarmi è così composta:

Tabella	Campo	Tipo dato	Info
alarms_data	machine↔_id	INT	
	id	INT	
	message	TEXT	

Questa tabella verrà aggiornata ciclicamente e conterrà solo gli allarmi presenti in quel momento sul macchinario

5.4.2 Tabella Ordini/Produzione

La tabella degli ordini e delle produzioni è così composta:

Tabella	Campo	Tipo dato	Info
orders	id	INT	
	machine_id	INT	
	order_id	TEXT	
	sequence↔_id	TEXT	
	batch_id	TEXT	
	article	TEXT	
	order_type	INT	Per il tipo di ordine i valori sono: 0 – Ordine solo con numeri di produzione 1 – Ordine con numeri di produzione e dimensioni <0 – Ordini manuali creati in macchina
	dimension↔_x	INT	
	dimension↔_y	INT	
	dimension↔_z	INT	
	items_todo	INT	
	items_done	INT	
	enabled	BOOLEAN	
	state	INT	
	custom_data	TEXT	
	order_data	DATE	
start_date	DATE		
end_date	DATE		

5.4.3 Tabella Eventi

La tabella degli eventi è così composta:

Tabella	Campo	Tipo dato	Info
events	machine_id	INT	
	id	INT	
	date	DATE	
	type	INT	
	custom_data	TEXT	
	message	TEXT	

5.4.4 Tabella Stati

La tabella degli stati è così composta:

Tabella	Campo	Tipo dato	Info
states_data	machine_id	INT	
	key (key_value per MSSql)	TEXT	
	value	TEXT	

5.4.5 Tabella statistiche

La tabella delle statistiche è così composta:

Tabella	Campo	Tipo dato	Info
statistics(statistics_data per MSSql)	machine↔ _id	INT	
	date	DATE	
	work_time	TEXT	
	stop_time	TEXT	
	quantity	INTEGER	

5.4.6 Tabella Extra dati degli Ordini

La tabella per i dati supplementari degli ordini è così composta:

Tabella	Campo	Tipo dato	Info
order_extra_data	machine↔ _id	INT	
	order_id	INT	
	data_name	TEXT	
	data_value	TEXT	

5.4.7 Tabella Ricette/Programmi

La tabella delle ricette/programmi è così composta:

Tabella	Campo	Tipo dato	Info
recipes	machine↔ _id	INT	
	value	TEXT	

Capitolo 6

Descrizione API Rest

Per poter richiedere i dati al server si possono utilizzare due metodi. Questi metodi sono:

- Comandi tramite browser WEB
- Utilizzo di chiamate WebRequest direttamente dai programmi

6.1 Richieste principali

I comandi per le richieste dati al server, devono essere inviate con questa sintassi:

```
<ip_server>:<porta_ip>/<richiesta>?cmd=<nome_id_comando> [&<nome_parametro>=<valore_parametro>]
```

Dove:

- **ip_server** – Indirizzo IP del server nella rete locale
- **porta_ip** – Numero della porta del server. È definita nel file di configurazione del programma
- **richiesta** – Nome della pagina da richiamare
- **nome_comando** – Nome del comando da gestire da parte della macchina o identificativo del dato da richiedere
- **nome_parametro** – Nome del parametro
- **valore_parametro** – Valore del parametro

Per i comandi che non devono restituire dati la risposta potrà contenere i seguenti valori:

Valore restituito	Descrizione
ok	L'operazione è andata a buon fine
<codice e descr>	Codice di errore con la descrizione associata

Di seguito vengono descritte le richieste nel dettaglio

6.1.1 Richiesta state

Richiesta per l'impostazione dei dati o la riletture dello stato corrente della macchina. Se non si imposta nessun comando, viene restituita una stringa nel formato scelto nel file di configurazione con i dati di tutti gli stati. I comandi disponibili sono:

- **set** – Comando per aggiungere o impostare un valore da memorizzare. I parametri per questo comando sono:
 - **name** – Nome dello stato
 - **value** – Valore dello stato
- **set_data** – Comando per aggiungere o impostare valori multipli da memorizzare. I parametri per questo comando sono:
 - Nome dello stato
 - Valore dello stato

Esempio per l'impostazione di dati con il comando **set**:

```
http://localhost:8010/state?cmd=set&name=V_STATO_TRP&value=0
```

Esempio per l'impostazione di dati con il comando **set_data**:

```
http://localhost:8010/state?cmd=set_data&V_STATO_TRP=0&V_ALLARMI=2&V_COUNT=12
```

Esempio per la richiesta degli stati:

```
http://localhost:8010/state
```

Risposta:

Tipo risposta	Dati risposta
XML	<pre><?xml version="1.0" encoding="UTF-8"?> <States> <State> <Name>V_STATO_TRP</Name> <Value>0</Value> </State> </States></pre>
JSON	<pre>{ "States": [{ "V_STATO_TRP": "0" }] }</pre>

6.1.2 Richiesta production

Richiesta per la gestione dei dati della produzione. Se non si imposta nessun comando, viene restituita una stringa nel formato scelto nel file di configurazione con i dati di tutti i record di produzione. I comandi disponibili sono:

- **add** – Comando per l'aggiunta di un record di produzione. I parametri per questo comando sono:
 - **id** – Identificativo della produzione. ATTENZIONE!! E' facoltativo, se non inviato viene sempre messo a -1
 - **name_or_id** – Nome del programma o ricetta per i dati di produzione
 - **quantity** – Quantità da produrre. ATTENZIONE!! Se il valore è con la virgola, mettere il carattere '.' Per la separazione dei decimali

- **custom_data** – Dati personalizzati. In questo dato, è possibile inserire anche gli altri dati di produzione seguendo la seguente sintassi:

```
<nome>:<valore>;<nome>:<valore>; ...
```

dove:

- * **nome** – Nome del dato da impostare
- * **valore** – Valore del dato da impostare. I nomi dei valori sono gli stessi dell'inserimento dati per il comando **orders**

Questo valore può contenere tutti i nomi dei parametri utilizzati per la richiesta **orders** perchè condivide la stessa tabella

- **remove** – Comando per l'eliminazione di un record di produzione. I parametri per questo comando sono:
 - **name_or_id** – Id del riferimento dei dati di produzione da eliminare
- **set_state** – Comando per l'impostazione dello stato di un record di produzione. I parametri per questo comando sono:
 - **name_or_id** – Id del riferimento dei dati di produzione da modificare
 - **state** – Nuovo stato da impostare. Gli stati definiti sono:
 - * **0** – In attesa
 - * **1** – In pausa
 - * **2** – In lavorazione
 - * **3** – Terminato
- **from_state** – Comando per la restituzione dei dati di produzione da un valore di stato. I parametri per questo comando sono:
 - **state** – Valore dello stato per la richiesta della lista dei records di produzione
- **set_todo** – Comando per l'impostazione della quantità di produzione. I parametri per questo comando sono:
 - **name_or_id** – id del riferimento dei dati di produzione da modificare
 - **num** – Nuovo valore di quantità
- **set_terminated** – Comando per l'impostazione della quantità prodotta. I parametri per questo comando sono:
 - **name_or_id** – id del riferimento dei dati di produzione da modificare
 - **num** – Nuovo valore di quantità
- **clear** – Comando per la cancellazione di tutti i records di produzione

Esempio per l'impostazione di dati di una produzione

```
http://localhost:8010/production?cmd=add&id=1&name_or_id=1&quantity=1000&custom_data=dimension_x:100;dimension_y:500
```

Esempio per l'eliminazione di una produzione

```
http://localhost:8010/production?cmd=remove&name_or_id=1
```

Esempio per la richiesta dei dati di produzione

```
http://localhost:8010/production
```

Risposta: Per il formato della risposta vedere la richiesta **orders**

6.1.3 Richiesta orders

Richiesta per la gestione dei dati degli ordini. Se non si imposta nessun comando, viene restituita una stringa nel formato scelto nel file di configurazione con i dati di tutti i record degli ordini. I comandi disponibili sono:

- **add** – Comando per l'aggiunta di un record ordine. I parametri per questo comando sono:
 - **id** – Identificativo della produzione. ATTENZIONE!! E' facoltativo, se non inviato viene sempre messo a -1
 - **order_id** – Testo per l'identificativo dell'ordine
 - **sequence_id** – Testo per l'identificativo della commessa
 - **batch_id** – Testo per l'identificativo del lotto
 - **order_type** – Tipo di ordine. Al momento non utilizzato
 - **name_or_id** – Nome del programma o ricetta per l'ordine
 - **quantity** – Quantità da produrre. ATTENZIONE!! Se il valore è con la virgola, mettere il carattere '.' Per la separazione dei decimali
 - **dimension_x** – Dimensione X dei pezzi da produrre
 - **dimension_y** – Dimensione Y dei pezzi da produrre
 - **dimension_z** – Dimensione Z dei pezzi da produrre
 - **start** – Data dell'ordine. ATTENZIONE!! Se non impostato viene assegnata la data corrente
 - **custom_data** – Dati personalizzati

- **remove** – Comando per l'eliminazione di un record di produzione. I parametri per questo comando sono:
 - **name_or_id** – Id del riferimento dei dati di produzione da eliminare

- **set_state** – Comando per l'impostazione dello stato di un record di produzione. I parametri per questo comando sono:
 - **name_or_id** – Id del riferimento dei dati di produzione da modificare
 - **state** – Nuovo stato da impostare. Gli stati definiti sono:
 - * **0** – In attesa
 - * **1** – In pausa
 - * **2** – In lavorazione
 - * **3** – Terminato

- **from_state** – Comando per la restituzione dei dati di produzione da un valore di stato. I parametri per questo comando sono:
 - **state** – Valore dello stato per la richiesta della lista dei records di produzione

- **set_todo** – Comando per l'impostazione della quantità di produzione. I parametri per questo comando sono:
 - **name_or_id** – id del riferimento dei dati di produzione da modificare
 - **num** – Nuovo valore di quantità

- **set_terminated** – Comando per l'impostazione della quantità prodotta. I parametri per questo comando sono:
 - **name_or_id** – id del riferimento dei dati di produzione da modificare
 - **num** – Nuovo valore di quantità

- **clear** – Comando per la cancellazione di tutti i records di produzione

Esempio per l'impostazione di dati di un ordine

http://localhost:8010/orders?cmd=add&id=1&order_id=122&name_or_id=ricetta1&quantity=1000&dimension_x=100&dimension_y=500

Esempio per l'eliminazione di un ordine

http://localhost:8010/orders?cmd=remove&name_or_id=1

Esempio per la richiesta dei dati di produzione

<http://localhost:8010/orders>

Risposta:

Tipo risposta	Dati risposta
XML	<pre><Orders> <Order> <Id>1</Id> <OrderId>122</OrderId> <SequenceId> </SequenceId> <BatchId> </BatchId> <Article>ricetta1</Article> <OrderType>0</OrderType> <DimensionX>1000</DimensionX> <DimensionY>500</DimensionY> <DimensionZ>0</DimensionZ> <QuantityTodo>1000</QuantityTodo> <QuantityTerminated>0</QuantityTerminated> <State>0</State> <CustomData>dimension_x:1000;dimension_y:500;order_id:122</CustomData> <OrderDate>10/10/2022 15:48:25</OrderDate> <StartDate>01/01/0001 00:00:00</StartDate> <EndDate>01/01/0001 00:00:00</EndDate> </Order> </Orders></pre>
JSON	<pre>{ "Orders": [{ "Id": "1", "OrderId": "122", "SequenceId": "", "BatchId": "", "Article": "ricetta1", "OrderType": 0, "DimensionX": 1000, "DimensionY": 500, "DimensionZ": 0, "QuantityTodo": 1000, "QuantityTerminated": 0, "State": 0, "Enabled": false, "CustomData": "dimension_x:1000;dimension_y:500;order_id:122", "OrderDate": "10\\10\\2022 15:48:25", "StartDate": "01\\01\\0001 00:00:00", "EndDate": "01\\01\\0001 00:00:00" }] }</pre>

nella tabella seguente, vengono descritti i dati dei valori restituiti dalla richiesta degli ordini

Nome	Descrizione
Id	È un identificativo se il dato di produzione dovesse essere prelevato da un database
OrderId	Id dell'ordine
SequenceId	Id della commessa
BatchId	Id della produzione
Article	Articolo da produrre
OrderType	Tipo di ordine
DimensionX	Dimensione X del pezzo
DimensionY	Dimensione Y del pezzo

Nome	Descrizione
DimensionZ	Dimensione Z del pezzo
QuantityTodo	Quantità da produrre
QuantityTerminated	Quantità prodotta
State	Stato della produzione. Per i valori vedere il comando <code>set_state</code> delle richieste per le produzioni.
Enabled	Flag di abilitazione della produzione, al momento non utilizzato
CustomData	Dati da mettere nelle note
OrderDate	Data dell'emissione dell'ordine
StartDate	Data di inizio lavorazione
EndDate	Data di fine lavorazione

6.1.4 Richiesta `order_extra_data`

Richiesta per la gestione dei dati extra degli ordini. Se non si imposta nessun comando, viene restituita una stringa nel formato scelto del file di configurazione con i dati di tutti i dati extra. I comandi disponibili sono:

- **set** – Comando per l'impostazione di un dato extra per un ordine. I parametri per questo comando sono:
 - **order_id** – ID dell'ordine associato al dato
 - **data_name** – Nome di riferimento del dato
 - **data_value** – Valore dell'extra dato
- **get** – Comando per la restituzione dei dati extra di un ordine. I parametri per questo comando sono:
 - **order_id** – ID dell'ordine associato al dato. **Se -1 restituisce tutti i dati**
- **clear** – Comando per la cancellazione dei dati extra di un ordine. I parametri per questo comando sono:
 - **order_id** – ID dell'ordine associato al dato. **Se -1 cancella tutti i dati**

Esempio per l'impostazione di dati con nome 'Priorita' a ordini diversi

```
http://localhost:8010/order_extra_data?cmd=set&order_id=1&data_name=Priorita&value=1
http://localhost:8010/order_extra_data?cmd=set&order_id=2&data_name=Priorita&value=8
```

Esempio per la cancellazione di tutti i dati extra degli ordini

```
http://localhost:8010/order_extra_data?cmd=clear
oppure
http://localhost:8010/order_extra_data?cmd=clear&order_id=-1
```

Esempio per la richiesta dei dati extra di tutti gli ordini:

```
http://localhost:8010/order_extra_data
oppure
http://localhost:8010/order_extra_data?order_id=-1
```

Risposta:

Tipo risposta	Dati risposta
XML	<pre><?xml version="1.0" encoding="UTF-8"?> <ExtraDatas> <ExtraData> <OrderId>1</OrderId> <DataName>Priorita</DataName> <DataValue>1</DataValue> </ExtraData> <ExtraData> <OrderId>2</OrderId> <DataName>Priorita</DataName> <DataValue>8</DataValue> </ExtraData> </ExtraDatas></pre>
JSON	<pre>{ "ExtraData": [{ "OrderId":1, "DataName":"Priorita", "DataValue":"1" }, { "OrderId":2, "DataName":"Priorita", "DataValue":"8" }] }</pre>

6.1.5 Richiesta events

Richiesta per la gestione degli eventi. Se non si imposta nessun comando, viene restituita una stringa nel formato scelto del file di configurazione con i dati di tutti gli eventi. I comandi disponibili sono:

- **add** – Comando per l’inserimento di un record evento. I parametri per questo comando sono:
 - **id** – Identificativo dell’evento
 - **subid** – identificativo secondario dell’evento. In pratica un ulteriore id per identificare cosa ha scatenato l’evento
 - **custom_data** – Dati custom dell’evento
 - **message** – Messaggio associato all’evento. Questo parametro è facoltativo
- **clear** – Comando per la cancellazione di tutti i records evento

Esempio per l’aggiunta di un evento

```
http://localhost:8010/events?cmd=add&id=100&subid=2&custom_data=dati extra&message=test di messaggio
```

Esempio per la cancellazione degli eventi

```
http://localhost:8010/events?cmd=clear
```

Esempio per la richiesta dei dati degli eventi del sistema

```
http://localhost:8010/events
```

Risposta:

Tipo risposta	Dati risposta
XML	<pre><?xml version="1.0" encoding="UTF-8"?> <Events> <Event> <Date>21/10/2021 21:51:57</Date> <Id>100</Id> <TypeData>2</TypeData> <CustomData>Dati extra</CustomData> <Message>test di messaggio</Message> </Event> </Events></pre>
JSON	<pre>{ "Events": [{ "Date": "21/10/2021 21:51:57", "Id": "12", "TypeData": "5", "CustomData": "", "Message": "test messaggio" }] }</pre>

6.1.6 Richiesta alarms

Richiesta per la gestione degli allarmi. Se non si imposta nessun comando, viene restituita una stringa nel formato scelto nel file di configurazione con i dati di tutti gli allarmi. I comandi disponibili sono:

- **add** – Comando per l’inserimento di un record allarme. I parametri per questo comando sono:
 - **id** – Identificativo dell’allarme
 - **message** – Messaggio associato all’allarme
- **clear** – Comando per la cancellazione di tutti i records allarme

Esempio per l’aggiunta di un allarme

<http://localhost:8010/alarms?cmd=add&id=2&message=Emergenza>

Esempio per la cancellazione degli allarmi

<http://localhost:8010/alarms?cmd=clear>

Esempio per la richiesta dei dati degli eventi del sistema

<http://localhost:8010/events>

Risposta:

Tipo risposta	Dati risposta
XML	<pre><?xml version="1.0" encoding="UTF-8"?> <Alarms> <Alarm> <Id>2</Id> <Message>Emergenza</Message> </Alarm> </Alarms></pre>
JSON	<pre>{ "Alarms": [{ "Id": 2, "Message": "Emergenza" }] }</pre>

6.1.7 Richiesta connection

Richiesta per il controllo dell'attività del server. Se viene utilizzata ed il server sarà attivo, risponderà con 'ok'. Utili per essere utilizzato come test e per il mantenimento della connessione

6.1.8 Richiesta statistics

Richiesta delle statistiche dello stato del macchinario. Queste statistiche vengono aggiornate direttamente dal macchinario. Se non si imposta nessun comando, viene restituita una stringa nel formato scelto nel file di configurazione con i dati delle statistiche salvate in quel momento. I comandi disponibili sono:

- **clear** – Comando per la cancellazione di tutti i records delle statistiche

Esempio per la cancellazione delle statistiche

```
http://localhost:8010/statistics?cmd=clear
```

Esempio per la richiesta dei dati delle statistiche

```
http://localhost:8010/statistics
```

Risposta:

Tipo risposta	Dati risposta
XML	<pre><?xml version="1.0" encoding="UTF-8"?> <Statistics> <Statistic> <Date>22/10/2021 16:59:03</Date> <WorkTime>00:00:00</WorkTime> <StopTime>00:02:53.2462568</StopTime> <Quantity>0</Quantity> </Statistic> </Statistics></pre>
JSON	<pre>{ "Statistics": [{ "current_time": "22/10/2021 16:59:03", "WorkTime": "00:00:00", "StopTime": "00:02:53.2462568", "Quantity": "0" }] }</pre>

6.1.9 Richiesta recipes

Richiesta di restituzione della lista delle ricette presenti in macchina. Non necessita dell'utilizzo di comandi o parametri. Viene restituita una stringa nel formato scelto nel file di configurazione con i nomi delle ricette della macchina

Esempio per la richiesta della lista delle ricette in macchina

```
http://localhost:8010/recipes
```

Risposta:

Tipo risposta	Dati risposta
XML	<pre><?xml version="1.0" encoding="UTF-8"?> <Recipes> <Recipe>Ricetta1</Recipe> <Recipe>Ricetta2</Recipe> </Recipes></pre>
JSON	<pre>{ "Recipes": ["Ricetta1", "Ricetta2"] }</pre>

6.1.10 Richiesta settings

Richiesta dei dati di configurazione utilizzabili dal client. Per la richiesta dei dati, come nome del comando, possono essere inseriti i seguenti valori:

- **MachineId** – Restituisce l'identificativo del macchinario
- **SerializerType** – Restituisce il tipo di serializzazione per i dati ricevuti. Al momento i tipi sono:
 - XML
 - JSON
- **PageLanguage** – Restituisce l'identificativo del linguaggio delle pagine HTML

Esempio per la richiesta dell'identificativo del macchinario

<http://localhost:8010/settings?cmd=MachineId>

Risposta: Restituisce una stringa con l'identificativo del macchinario

Capitolo 7

Storico delle versioni

Qui di seguito vengono descritti gli storici delle versioni del framework e del documento.

7.1 Storico framework

Revisione	Data	Descrizione
V1.0.0.0	01/12/2020	Prima versione
V1.0.0.1	13/01/2021	Aggiunta la gestione dell'id del macchinario per il database
V1.0.0.6	15/01/2021	Aggiunta la gestione dei dati criptati per user e password per l'accesso al server ed al database
		Aggiunta la gestione del log
V1.0.0.7	21/01/2021	Gestione dei dati custom per gli eventi
V1.0.0.8	02/04/2021	Gestione a plugins per i gestori dati
V1.0.0.9	28/04/2021	Controllo di presenza applicazione all'avvio del programma
		Preparazione per il caricamento dei server come plugins
V1.0.1.0	07/05/2021	Aggiunto gestore DB MySQL
		Inizio gestione per backup
V1.0.1.1	23/07/2021	Aggiunta registrazione del percorso completo dell'eseguibile nel registro
V1.0.1.2	05/08/2021	Modifiche per l'uso della nuova libreria PxIndustry40.SystemDevice
		Gestione per la scrittura di variabili dal gestore device
V1.0.1.3	08/09/2021	Aggiunto un campo al file di configurazione per il file dei messaggi degli eventi
V1.0.1.4	14/09/2021	Aggiunto un campo 'id' per l'inserimento di un dato di produzione con identificativo
V1.0.1.5	17/09/2021	Aggiunto handler per la richiesta dei dati delle ricette al server
V1.0.1.6	21/09/2021	Aggiunta la possibilità di selezionare il gestore della serializzazione dei dati da restituiti file di configurazione. Al momento serializza solo in XML
V1.0.1.7	06/10/2021	Aggiunto il valore di risoluzione nel file di configurazione per la conversione dei valori delle variabili
		Correzione bug per la restituzione dei dati custom delle produzioni
V1.0.1.8	16/10/2021	Terminata serializzazione con JSON. Impostabile dalla configurazione
V1.0.1.9	22/10/2021	Modifiche a plugins DB dopo test di funzionamento globale
V1.0.2.0	03/11/2021	Aggiunto il gestore PROXY_SYS per la lettura dei dati del device tramite connessione di rete per collegarsi al server avviato dal programma del pannello e leggere i dati dalla stessa connessione al controllo (Per NGMEVO)
V1.0.2.2	23/05/2022	Modifiche per l'aggiornamento della tabella degli ordini
V2.0.0.0	12/10/2022	Nuova gestione con libreria da integrare nei vari software senza la struttura a plugin

Revisione	Data	Descrizione
		Aggiunti controlli Form e WPF nel namespace Promax.Industry40.UI' per la gestione dei dati sui pannelli dei macchinari
V2.0.0.1	27/10/2022	Aggiunta la chiave 'ViewCustomColumn' per la visualizzazione di un campo inserito nelle note nei controlli UI presenti in libreria
V2.0.0.2	22/12/2022	Gestione delle variabili puntatore per la lettura direttamente dalla memoria del controllo. Devono essere variabili di tipo LONG (4 byte)
V2.0.0.3	26/01/2023	Inizializzazione della dictionary dei servizi in ogni metodo di Px40Services per evitare eccezioni
V2.0.0.4	04/02/2023	Centralizzazione in Px40Services della proprietà NI per la formattazione dei dati della libreria
		Inizio Scrittura commenti per la documentazione
V2.0.0.5	13/02/2023	Aggiunto un metodo per la restituzione della lista dati delle statistiche
V2.0.0.6	23/02/2023	Aggiunto l'attributo 'OffsetFromBaseAddress' nella configurazione delle variabili per applicare uno scostamento all'indirizzo base
V2.0.0.7	25/02/2023	Aggiunta la voce di configurazione 'FlagsEnableLog' per l'abilitazione dei tipi di logs (Default = 0)
		Modificata la gestione dei dati DBStatistics
V2.0.0.8	12/03/2023	Modificata la gestione dei conteggi delle statistiche. E' stato aggiunto il dato di produzione giornaliera
V2.0.0.9	03/05/2023	Aggiunto un controlloWPF per la visualizzazione degli eventi
		Aggiunta la gestione delle traduzioni in WPF tramite markup
		Aggiunto un costruttore per i dati DBOrderItem per la clonazione dell'oggetto
		Aggiunta un metodo per la richiesta degli eventi entro un range di dati
		Aggiunto un metodo per la restituzione del percorso completo di una ricetta
		Aggiunto un flag nelle impostazioni per l'uso del percorso assoluto della cartella ricette
V2.0.0.1	05/05/2023	Aggiunta la gestione degli stili dei controlli WPF dall'applicazione host se registrata l'interfaccia IPxUIService
V2.0.0.11	09/05/2023	Modifiche per la gestione del cambio di stato di un ordine. Ora prima di impostare il nuovo ordine viene richiesta la quantità dell'ordine corrente
		Modifiche per l'inserimento della quantità giornaliera nelle statistiche
		Modifiche per l'aggiunta, la modifica e l'eliminazione di un rdine in locale
		Aggiunto il tipo DEBUG per i tipi di logs
		Aggiunto un metodo per la restituzione della lista delle ricette
		Modifiche per la lettura dalla cartella base delle sottocartelle durante la richiesta delle ricette
V2.0.0.12	16/05/2023	Aggiunta una form per i dati di PlantDCS (Solo per la versione Full)
V2.0.0.13	19/05/2023	Aggiunto un handler 'popup' per la trasmissione di messaggi al server
V2.0.0.14	22/05/2023	Aggiunta la gestione dei comandi nell'handler 'popup'
V2.0.0.15	16/06/2023	Aggiunti metodi per la copia e la clonatura dei dati dell'ordine
V2.0.0.16	04/08/2023	Eliminato il progetto EPPlus per l'uso della sola libreria
		Aggiunto un metodo per l'aggiornamento dei dati delle statistiche senza inserire i valori delle date
		Aggiunti i files della documentazione
V2.0.0.17	24/11/2023	Aggiunta una tabella DB per i dati delle ricette del macchinario
		Aggiunta una tabella DB per gli extra dati di un ordine
V2.0.0.18	01/12/2023	Letture dei dati delle ricette da DB se il nome della cartella sorgente non è disponibile. La lista delle ricette deve essere aggiornata dall'applicazione Host

7.2 Storico documentazione

Revisione	Data	Descrizione
1.0	07/01/2021	Versione iniziale
1.1	13/01/2021	Aggiunta gestione dell'ID della macchina
1.2	07/04/2021	Aggiunti comandi per le produzioni
		Aggiunta descrizione utilizzo plugins per gestore dati
1.3	04/06/2021	Aggiunta descrizione del comando 'chat'
1.4	13/07/2021	Aggiustamento struttura capitoli
		Aggiunta la descrizione dei plugins del sistema
1.5	03/08/2021	Aggiunti attributi per le variabili del file di configurazione
		Aggiunte altre voci per i parametri Settings
		Aggiunta di nomi predefiniti per le variabili del file di configurazione
1.6	01/09/2021	Aggiunta una tabella per i dati degli stati
1.7	08/09/2021	Aggiunga la descrizione del file per i messaggi degli eventi
1.8	16/10/2021	Allineate le chiavi delle impostazioni del file di configurazione
		Aggiunta la descrizione per l'uso di dati JSON da codice Javascript
1.9	22/10/2021	Aggiunta la descrizione dei dati di risposta con i vari tipi di serializzazione
2.0	07/03/2022	Aggiunto capitolo per la struttura e le fasi di avvio del software di gestione del macchinario
2.1	09/04/2022	Aggiunta la descrizione per gestore dati SQLite
2.2	17/06/2022	Aggiunta la lista dei dati delle tabelle DB
2.3	21/06/2022	Aggiunte le descrizioni per la configurazione dell'accesso da remoto dei servers DB
2.4	16/09/2022	Aggiornamento per la descrizione delle tabelle DB
2.5	25/10/2022	Passaggio alla versione 2.0.0.0 per la separazione delle librerie
2.6	27/10/2022	Aggiunta la chiave 'ViewCustomColumn' per la configurazione
2.7	07/11/2022	Aggiunto l'attributo 'IsVirtual' nelle definizioni di una variabile
		Aggiunto l'attributo 'QuantityResolution' per la risoluzione delle quantità
2.8	16/12/2022	Aggiunta la chiave 'ViewSortingCriterion' per l'ordinamento della lista degli ordini nelle finestre dei dati
		Aggiunta la chiave 'ViewHideColumns' per nascondere le colonne degli ordini nelle finestre dei dati
2.9	23/02/2023	Aggiunto l'attributo 'OffsetFromBaseAddress' alle variabili
		Aggiunta la chiave 'FlagsEnableLog' nella configurazione
3.0	05/05/2023	Aggiunta la chiave 'RecipesPathIsAbsolute'
3.1	22/05/2023	Aggiunto un comando per l'invio di messaggi da parte dei client
3.2	24/11/2023	Aggiunta una tabella 'recipes' per la lista di ricette/programmi presenti nel macchinario
		Aggiunta una tabella 'order_extra_data' per i dati extra associati all'ordine
		Aggiunto il comando 'order_extra_data' per la restituzione dei dati extra di un ordine

Indice analitico

Descrizione API Rest, [25](#)

File dei dati degli eventi, [15](#)

File di configurazione, [9](#)

Gestione dei database, [17](#)

Integrazione delle librerie, [3](#)

Introduzione, [1](#)

Storico delle versioni, [35](#)