

Bem-vindo à Eztech!

Aqui, você integra de forma rápida, simples e segura. Pensamos em cada passo para que você tenha uma experiência única e consiga integrar com poucas linhas de código.

Nesta documentação você pode explorar as principais funcionalidades que a nossa API oferece e adaptá-la ao seu sistema.

Documentação

Nesta documentação apresentamos o passo a passo para integrar seu sistema com os sistemas da EZTech. A nossa API de exemplo é um software que contém as principais funcionalidades utilizadas no dia a dia de um posto de combustível. Aqui você verá como utilizar essa API.

******Caso queira integrar com a medição de tanque é só entrar em contato com a EZTech e solicitar o Quick manual (pode enviar um e-mail para desenvolvimento@eztech.ind.br).

Sumário

Autentificação	4
Abastecimento	6
Bombas	9
Leitura do cartão	11
Autorizar bomba	12
Bloquear bomba	13
Ver preço	13
Troca de preço	14
Predeterminação(Preset)	17
Finalizar abastecimento	18
Desativar bico	18
Consultar entregas	19
Lista de sensores	22
Lista dos ZigBees	23
Lista dos bicos	24
Lista dos encerrantes	25
Processamento de eventos	26
Eventos da bomba	28
Eventos da abastecimento	31
Eventos de leitura dos cartões	34
Eventos de mudança de encerrantes	35
Eventos do EZServer	36
Eventos do Client	37
Eventos dos dispositivos ZigBee	37
Eventos de Log	38

Autentificação

Descrição: O processo de autentificação é bastante simples. No processo você pode optar por trabalhar por eventos ou sempre conectado ao concentrador. Abaixo você consegue visualizar o procedimento necessário para se conectar ao nosso concentrador:

Exemplo:

```
private void btLogon_Click(object sender, EventArgs e)
{
    short tipoDeCliente;
    int clienteID = 35;
    IntPtr iprt = new IntPtr(0);
    DateTime dateTime = DateTime.Now;

    // Identificação de como você irá trabalhar, se será por
    // eventos ou se você ficará sempre conectado.
    if (chProcEvents.Checked)
        tipoDeCliente = 7;
    else
        tipoDeCliente = 1;

    if (edServerAddress.Enabled == true)
    {
        WriteMessage("Conectando no servidor: " + edServerAddress.Text);

        if (GoodResult(EZInterface.ClientLogonEx(clienteID, tipoDeCliente,
            edServerAddress.Text, 5123, 5124, 10000, 0, new IntPtr(0), 0)))
        {
            edServerAddress.Enabled = false;
            chProcEvents.Enabled = false;
            btLogon.Text = "Logoff";

            EZInterface.SetClientType(EZInterface.SINK_CLIENT_EVENT);
        }
    }
}
```

```
        if (GoodResult(EZInterface.SetDateTime(dateTime)))  
            WriteMessage("Data e Hora do concentrador atualizada com  
sucesso");  
        }  
        else  
        {  
            WriteMessage("Desconectando do servidor: " +  
edServerAddress.Text);  
            GoodResult(EZInterface.ClientLogoff());  
            edServerAddress.Enabled = true;  
            chProcEvents.Enabled = true;  
            btLogon.Text = "Logon";  
        }  
    }  
}
```

Abastecimento

Descrição: A EZTech disponibiliza mais de uma forma de coleta dos abastecimentos. Com o intuito de facilitar o seu trabalho vou mencionar a principal e no final deste documento estará a forma de coleta por eventos.

Exemplo:

```
private void btGetAllDeliveries_Click(object sender, EventArgs e)
{
    int contador = 0;
    int quantidadeDeAbastecimentos = 0;
    int Id = 0;
    int HoseID = 0;
    short State = 0;
    short DType = 0;
    double Volume = 0;
    short PriceLevel = 0;
    double Price = 0;
    double Value = 0;
    double Volume2 = 0;
    DateTime CompletedDT = new DateTime();
    int LockedBy = 0;
    int ReservedBy = 0;
    int AttendantID = 0;
    int Age = 0;
    DateTime ClearedDT = new DateTime();
    double OldVolumeETot = 0;
    double OldVolume2ETot = 0;
    double OldValueETot = 0;
    double NewVolumeETot = 0;
    double NewVolume2ETot = 0;
    double NewValueETot = 0;
    Int64 Tag = 0;
    int Duration = 0;
    int ClientID = 0;
```

```
// Verifica se esta conectado ao servidor
if (EZInterface.TestConnection() != 0)
    return;

// Le o numero de abastecimentos que estão no Ezserver, ou seja, não pegos
por nenhum Client.
if (!GoodResult(EZInterface.GetDeliveriesCount(ref
quantidadeDeAbastecimentos)))
    return;
if (quantidadeDeAbastecimentos != 0)
{
    WriteMessage("[Abastecimentos " + quantidadeDeAbastecimentos +
        "]------");
    for (contador = quantidadeDeAbastecimentos; contador > 0; contador--)
    {
        if (!GoodResult(EZInterface.GetDeliveryByOrdinal(contador, ref Id)))
            return;
        if (GoodResult(EZInterface.GetDeliveryPropertiesEx3(Id, ref HoseID,
            ref State, ref DType, ref Volume, ref PriceLevel, ref Price, ref Value,
            ref Volume2, ref CompletedDT, ref LockedBy, ref ReservedBy,
            ref AttendantID, ref Age, ref ClearedDT, ref OldVolumeETot,
            ref OldVolume2ETot, ref OldValueETot, ref NewVolumeETot,
            ref NewVolume2ETot, ref NewValueETot, ref Tag, ref Duration,
            ref ClientID)))
        {
            WriteMessage("----- Abastecimento: (" + contador + ") " + Id);
            WriteMessage("      HoseID " + HoseID + ", State " + State +
                ", Type " + DType);
            WriteMessage("      Volume " + Volume +
                ", PriceLevel " + PriceLevel + ", Price " + Price +
                ", Value " + Value);
            WriteMessage("      Volume2 " + Volume2 +
                ", CompletedDT " + CompletedDT + ", LockedBy " + LockedBy
                + ", ReservedBy " + ReservedBy);
        }
    }
}
```

```
WriteMessage("    AttendantID " + AttendantID + ", Age " + Age +  
    ", ClearedDT " + ClearedDT);  
WriteMessage("    OldVolumeETot " + OldVolumeETot +  
    ", OldVolume2ETot " + OldVolume2ETot +  
    ", OldvalueETot " + OldValueETot);  
WriteMessage("    NewVolumeETot " + NewVolumeETot +  
    ", NewVolume2ETot " + NewVolume2ETot +  
    ", NewValueETot " + NewValueETot);  
WriteMessage("    Tag " + Tag + ", Duraction " + Duration +  
    ", ClientID " + ClientID);  
WriteMessage("");  
if (LockedBy != -1)  
    continue;  
if (GoodResult(EZInterface.LockDelivery(Id)))  
    LockedBy = 1;  
else  
    continue;  
if ((LockedBy == 1) && (State !=  
(short)EZInterface.TDeliveryState.CLEARED))  
    GoodResult(EZInterface.ClearDelivery(Id, DType));  
}  
}  
}  
else  
{  
    WriteMessage("Sem abastecimentos até o momento.");  
}  
WriteMessage("-----");  
}
```

Bombas

Descrição: A EZTech disponibiliza uma forma para leitura dos status da bomba. Abaixo você pode ver um exemplo para captura desses status. Este método é chamado em um loop interno sempre apresentando os status da bomba.

Exemplo:

```
private void ReadPumpsStatus()
{
    int PumpsCount = 0;
    String PumpStates = "";
    String CurrentHose = "";
    String DeliveriesCount = "";
    int Idx = 0;
    int CurStatus = 0;
    int CurHose = 0;
    int CurDelv = 0;
    String StrStatus = "";
    byte[] cstatus;
    byte[] chose;
    byte[] cdeliv;
    System.Text.ASCIIEncoding conv = new System.Text.ASCIIEncoding();
    // Verifica se está conectado ao servidor
    if (EZInterface.TestConnection() == 0)
    {
        // Verifica a quantidade de bombas configuradas
        if (!GoodResult(EZInterface.GetPumpsCount(ref PumpsCount)))
            return;
        // Lê o estado de todas as bombas configuradas
        if (!GoodResult(EZInterface.GetAllPumpStatuses(ref PumpStates, ref
            CurrentHose, ref DeliveriesCount)))
            return;
        cstatus = conv.GetBytes(PumpStates);
        chose = conv.GetBytes(CurrentHose);
        cdeliv = conv.GetBytes(DeliveriesCount);
        for (Idx = 1; Idx <= PumpsCount; Idx++)
        {
            CurStatus = cstatus[Idx - 1] - '0'; //
            EZClient.TPumpState(Ord(PumpStates[Idx]) - Ord('0'));
            CurHose = chose[Idx - 1] - '0';
            CurDelv = cdeliv[Idx - 1] - '0';
            switch ((EZInterface.TPumpState)CurStatus)
            // PAM10)
            {
                case EZInterface.TPumpState.INVALID_PUMP_STATE:
                    StrStatus = "estado invalido."; break; // 0 - OFFLINE
```



```
        case EZInterface.TPumpState.NOT_INSTALLED_PUMP_STATE:
StrStatus = "nao instalada."; break; // 6 - CLOSE
        case EZInterface.TPumpState.NOT_RESPONDING_1_PUMP_STATE:
StrStatus = "Bomba nao responde."; break; // 0 - OFFLINE
        case EZInterface.TPumpState.IDLE_PUMP_STATE:
StrStatus = "em espera (desocupada)."; break; // 1 - IDLE
        case EZInterface.TPumpState.PRICE_CHANGE_STATE:
StrStatus = "troca de preco."; break; // 1 - IDLE
        case EZInterface.TPumpState.AUTHED_PUMP_STATE:
StrStatus = "Bomba Autorizada"; break; // 9 - AUTHORIZED
        case EZInterface.TPumpState.CALLING_PUMP_STATE:
StrStatus = "esperando autorizacao."; break; // 5 - CALL
        case EZInterface.TPumpState.DELIVERY_STARTING_PUMP_STATE:
StrStatus = "abastecimeneto iniciando."; break; // 2 - BUSY
        case EZInterface.TPumpState.DELIVERING_PUMP_STATE:
StrStatus = "abastecendo."; break; // 2 - BUSY
        case EZInterface.TPumpState.TEMP_STOPPED_PUMP_STATE:
StrStatus = "parada temporaria (no meio de uma abastecimento) (STOP)."; break;
// 8 - STOP
        case EZInterface.TPumpState.DELIVERY_FINISHING_PUMP_STATE:
StrStatus = "abastecimento finalizando (fluxo de produto diminuindo)."; break;
// 2 - BUSY
        case EZInterface.TPumpState.DELIVERY_FINISHED_PUMP_STATE:
StrStatus = "abastecimento finalizado (parou de sair combustivel)."; break;
// 2 - BUSY
        case EZInterface.TPumpState.DELIVERY_TIMEOUT_PUMP_STATE:
StrStatus = "abastecimento excedeu tempo maximo."; break; // 1 - IDLE
        case EZInterface.TPumpState.HOSE_OUT_PUMP_STATE:
StrStatus = "bico fora do guarda-bico (CALL)."; break; // 5 - CALL
        case EZInterface.TPumpState.PREPAY_REFUND_TIMEOUT_STATE:
StrStatus = "prazo de pre-determinacao esgotado."; break; // 1 - IDLE
        case EZInterface.TPumpState.DELIVERY_TERMINATED_STATE:
StrStatus = "abastecimento terminado (EOT)"; break; // 3 - EOT
        case EZInterface.TPumpState.ERROR_PUMP_STATE:
StrStatus = "Erro (resposta de erro da bomba)."; break; // 0 - OFFLINE
        case EZInterface.TPumpState.NOT_RESPONDING_2_PUMP_STATE:
StrStatus = "EZID nao responde."; break;
        case EZInterface.TPumpState.LAST_PUMP_STATE:
StrStatus = "Ultimo estado da bomba?"; break;
        default:
StrStatus = "estado desconhecido = " + CurStatus;
break;
    }
```

```
        if (lastStates[Idx - 1] != cstatus[Idx - 1])
        {
            WriteMessage("Bomba " + Idx + ", Bico " + chose[Idx - 1] +
                ", Pendentes " + cdeliv[Idx - 1] +
                ", Status: " + StrStatus);
            //lastStates.IndexOf(PumpStates[Idx], Idx);
            lastStates[Idx - 1] = cstatus[Idx - 1];
        }
    }
}
```

Leitura do cartão

Descrição: Método para leitura do cartão.

Exemplo:

```
private void btReadCards_Click(object sender, EventArgs e)
{
    int CardID = 0;
    int Number = 0;
    string Name = "";
    int PumpID = 0;
    short CardType = 0;
    int ParentID = 0;
    Int64 Tag = 0;
    DateTime TimeStamp = new DateTime();

    // Verifica conexao
    if (!GoodResult(EZInterface.TestConnection()))
        return;

    while (true)
    {
        // Le ID do primeiro cartao da lista
        if (EZInterface.GetCardReadByOrdinal(1, ref CardID) != 0)
            break;

        // LE dados do cartao
        if (!GoodResult(EZInterface.GetCardReadProperties(CardID, ref Number,
            ref Name, ref PumpID, ref CardType, ref ParentID, ref Tag, ref TimeStamp)))
            break;

        WriteMessage(" --- Cartao Lido: " +
            " ID= " + CardID +
            ", Numero= " + Number +
            ", Nome= " + Name +
```

```
        ", PumpID= " + PumpID +  
        ", CardType= " + CardType +  
        ", ParentID= " + ParentID +  
        ", Tag= " + Tag.ToString("X10") +  
        ", TimeStamp= " + TimeStamp);  
  
        // Esta função só funciona se a Autorizacao estiver configurada com um tipo de  
        // cartão:  
        // "Carta/Placa", "Frentista", "Cliente", "Frentista E Cliente", "Frentista OU Cliente"  
        if (GoodResult(EZInterface.TagAuthorise(PumpID, Tag,  
(short)EZInterface.TAllocLimitType.NO_LIMIT_TYPE, 0, 0xFF, 1)))  
            break;  
  
        WriteMessage(" --- Bomba " + PumpID + " Autorizada com Cartao " + Tag);  
  
        // Apaga Cartao da lista  
        if (!GoodResult(EZInterface.DeleteCardRead(CardID)))  
            break;  
    }  
}
```

Autorizar bomba

Descrição: Método para autorizar a bomba.

Exemplo:

```
private void btAuthorize_Click(object sender, EventArgs e)  
{  
    int Bomba = 0;  
    int IdBomba = 0;  
    Bomba = cbPump.SelectedIndex + 1; // Le o numero da bomba  
  
    // Verifica conexao  
    if (!GoodResult(EZInterface.TestConnection()))  
        return;  
  
    // Pega Id da Bomba escolhida  
    if (!GoodResult(EZInterface.GetPumpByOrdinal(Bomba, ref IdBomba)))  
        return;  
  
    // Envia Autorizacao para bomba  
    if (GoodResult(EZInterface.Authorise(IdBomba)))  
        WriteMessage(" --- Bomba " + Bomba + " Autorizada!");  
}
```

Bloquear bomba

Descrição: Método para bloquear a bomba.

Exemplo:

```
private void btLock_Click(object sender, EventArgs e)
{
    int Bomba = 0;
    int IdBomba = 0;
    Bomba = cbPump.SelectedIndex + 1; // Le o numero da bomba

    // Verifica conexao
    if (!GoodResult(EZInterface.TestConnection()))
        return;

    // Pega Id da Bomba escolhida
    if (!GoodResult(EZInterface.GetPumpByOrdinal(Bomba, ref IdBomba)))
        return;

    // Envia bloqueio (desautorizacao) para bomba
    if (GoodResult(EZInterface.CancelAuthorise(IdBomba)))
        WriteMessage("--- Bomba " + Bomba + " Desautorizada!");
}
```

Ver preço

Descrição:

Exemplo:

```
private void buttonSeePrice_Click(object sender, EventArgs e)
{
    int Id = 1;
    int Ct = 0;
    int Number = 0;
    int PumpID = 0;
    int TankID = 0;
    int PhysicalNumber = 0;
    double MtrTheoValue = 0;
    double MtrTheoVolume = 0;
    double MtrElecValue = 0;
    double MtrElecVolume = 0;
    short UVEAntenna = 0;
    double Price1 = 0;
    double Price2 = 0;
    short Enabled = 0;
    WriteMessage("<<<<-----GetHoseProperties ----->>>>");
}
```

```
if (!GoodResult(EZInterface.GetHosesCount(ref Ct)))  
    return;  
  
for (Id = 1; Id <= Ct; Id++)  
{  
    if (GoodResult(EZInterface.GetHosePropertiesEx2(Id, ref Number, ref PumpID,  
        ref TankID, ref PhysicalNumber, ref MtrTheoValue,  
        ref MtrTheoVolume, ref MtrElecValue, ref MtrElecVolume,  
        ref UVEAntenna, ref Price1, ref Price2, ref Enabled)))  
    {  
        WriteMessage("    ID: " + Id + ", Bico: " + Number + ", PumpID: " + PumpID +  
            ", TankID: " + TankID);  
        WriteMessage("        MtrTheoValue: " + MtrTheoValue +  
            ", MtrTheoVolume: " + MtrTheoVolume +  
            ", PhysicalNumber: " + PhysicalNumber);  
        WriteMessage("        MtrElecValue: " + MtrElecValue +  
            ", MtrElecVolume: " + MtrElecVolume);  
        WriteMessage("        UVEAntena: " + UVEAntenna + ", Price1: R$" + Price1 +  
            ", Price2: R$" + Price2 + ", Enables: " + Enabled);  
    }  
}  
}
```

Troca de preço

Descrição: Neste método, segundo o projeto exemplo, você informa o valor que deseja substituir o atual. O método pega esse valor e faz a troca internamente, salvando também no EZServer.

Exemplo:

```
private void btChangePrice_Click(object sender, EventArgs e)  
{  
    int Bomba = 0;  
    int Bico = 0;  
    int IdBico = 0;  
    short Duracao = 0;  
    short Tipo = 0;  
    double Valor1 = 0;  
    double Valor2 = 0;  
    int Index = 0;  
    int Bicos = 0;  
    int HNumber = 0;  
    int PhysicalNumber = 0;  
    int PumpID = 0;  
    int PumpNumber = 0;  
    int TankID = 0;
```

```
int TankNumber = 0;
int GradeID = 0;
int GradeNumber = 0;
double MtrTheoValue = 0;
double MtrTheoVolume = 0;
double trElecValue = 0;
double MtrElecVolume = 0;
double Price1 = 0;
double Price2 = 0;
short HEnabled = 0;
String PumpName = "";
String TankName = "";
String GradeName = "";
String GradeShortName = "";
String GradeCode = "";

// Duracao do preco (Multipos abastecimentos)
Duracao = (short)EZInterface.TDurationType.MULTIPLE_DURATION_TYPE;

// Tipo de preco (Fixo)
Tipo = (short)EZInterface.TPriceType.FIXED_PRICE_TYPE;

// Le o numero da bomba
Bomba = cbPump.SelectedIndex + 1;

// Le o numero do bico
Bico = cbHose.SelectedIndex + 1;

// Le o valor 1
Valor1 = Convert.ToDouble(edPrice1.Text);

// Le o valor 2
Valor2 = Convert.ToDouble(edPrice2.Text);
if (Valor2 == 0)
{
    Valor2 = Valor1;
}
WriteMessage("--- Bomba " + Bomba + " - Troca de precos");

// Verifica conexao
if (!GoodResult(EZInterface.TestConnection()))
    return;
// Le o numero de bicos cadastrados
if (!GoodResult(EZInterface.GetHosesCount(ref Bicos)))
    return;
for (Index = 1; Index <= Bicos; Index++)
```

```
{
    // Pega o ID do bico
    if (!GoodResult(EZInterface.GetHoseByOrdinal(Index, ref IdBico)))
        return;

    // Pega os dados do bico
    if (GoodResult(EZInterface.GetHoseSummaryEx(IdBico, ref HNumber,
        ref PhysicalNumber, ref PumpID, ref PumpNumber, ref PumpName,
        ref TankID, ref TankNumber, ref TankName, ref GradeID, ref GradeNumber,
        ref GradeName, ref GradeShortName, ref GradeCode, ref MtrTheoValue,
        ref MtrTheoVolume, ref trElecValue, ref MtrElecVolume, ref Price1,
        ref Price2, ref HEnabled)))
    {
        // Verifica se o ID do bico pertence ao escolhido
        if ((Bomba == PumpNumber) && (Bico == HNumber))
        {
            WriteMessage("      Precos Atual: Bomba " + PumpNumber +
                ", Bico " + HNumber + ", Preco1 R$" + Price1 +
                ", Preco2 R$" + Price2);
            if (GoodResult(EZInterface.SetHosePrices(IdBico, Duracao, Tipo, Valor1,
                Valor2)))
            {
                WriteMessage("      Preco Novo: Bomba " + Bomba + ", Bico " + Bico +
                    ", Preco1 R$" + Valor1 + ", Preco2 R$" + Valor2 +
                    ", (Duracao " + Duracao + ", Tipo " + Tipo + ")");
            }
            break;
        }
    }
}
```

Predeterminação(Preset)

Descrição: Neste método é possível criar um preset para valor ou volume.

Exemplo:

```
private void btPreset_Click(object sender, EventArgs e)
{
    int Bomba = 0;
    int Bico = 0;
    int IdBomba = 0;
    int IdBico = 0;
    short LType = 0;
    double PsValue = 0;

    // Le o numero da bomba
    Bomba = cbPump.SelectedIndex + 1;

    // Le o numero do bico
    Bico = cbHose.SelectedIndex + 1;

    LType = (short)(cbPresetType.SelectedIndex + 2);

    // Verifica se o textbox de Predet. está vazio
    if (edPreset.MaskCompleted)
    {
        PsValue = Convert.ToDouble(edPreset.Text);
    }
    else
    {
        WriteMessage("Valor informado está incorreto.");
    }

    WriteMessage("--- Bomba " + Bomba + " - Preset");

    // Verifica conexao
    if (!GoodResult(EZInterface.TestConnection()))
        return;

    // Pega Id da Bomba escolhida
    if (!GoodResult(EZInterface.GetPumpByOrdinal(Bomba, ref IdBomba)))
        return;

    IdBico = (1 << (Bico - 1)); // Calcula ID do bico escolhido

    // Envia preset para bomba
    if (GoodResult(EZInterface.LoadPreset(IdBomba, LType, PsValue, (short)IdBico, 1)))
        WriteMessage("    Preset Enviado: Bomba " + Bomba + " Bico " + Bico +
            " Tipo " + LType + " Valor " + PsValue + " Nivel 1");
}
```


Finalizar abastecimento

Descrição: Método para finalizar um abastecimento.

Exemplo:

```
private void button1_Click(object sender, EventArgs e)
{
    int Bomba;
    Bomba = cbPump.SelectedIndex + 1;
    EZInterface.TempStop(Bomba);
}
```

Desativar bico

Descrição: Método para desativar um bico.

Exemplo:

```
private void buttonDesativar_Click(object sender, EventArgs e)
{
    int Bomba = 0;
    int IdBomba = 0;
    int num = 0;

    // Le o numero da bomba
    Bomba = cbPump.SelectedIndex + 1;

    // Verifica conexao
    if (!GoodResult(EZInterface.TestConnection()))
        return;

    if (!GoodResult(EZInterface.GetHoseByOrdinal(Bomba, ref IdBomba)))
        return;
    if (!GoodResult(EZInterface.DisablePump(IdBomba)))
        return;
    if (!GoodResult(EZInterface.GetHosesCount(ref num)))
        return;

    //1 Bomba com o Id 1 = total de 4 bicos
    WriteMessage("Quantidade de bicos " + num);
}
```

Consultar entregas

Descrição: Método para consultar todas as entregas pendentes.

Exemplo:

```
private void buttonConsultarEntrega_Click(object sender, EventArgs e)
{
    Int32 Count = 0;
    Int16 DeviceType = 3; //TANK_ALR
    Int32 DeviceId = -1; //NULL_ID - Pegar todos os valores
    Int32 DeviceNumber = 0;
    string DeviceName = "";
    string EventDesc = "";
    Int16 EventLevel = -1;
    Int16 EventType = -1;
    Int32 ClearedBy = -2;
    Int32 AckedBy = -2;
    DateTime GeneratedDT = new DateTime();
    DateTime ClearedDT = new DateTime();
    double Volume = 0;
    double Value = 0;
    double ProductVolume = 0;
    double ProductLevel = 0;
    double WaterLevel = 0;
    double Temperature = 0;
    int Id = 0;

    //Testando a conexão
    if (GoodResult(EZInterface.TestConnection()))
    {
        if (EZInterface.GetLogEventCount(ref Count, DeviceType, DeviceId, EventLevel,
            EventType, ClearedBy, AckedBy) != 0)
            return;

        WriteMessage("[Eventos " + Count + "]-----");

        //Vem com o Count preenchido corretamente de acordo com o filtro que passei.
        for (int Index = 1; Index <= Count; Index++)
        {
            if (EZInterface.GetLogEventByOrdinal(Index, ref Id, DeviceType, DeviceId,
                EventLevel, EventType, ClearedBy, AckedBy) != 0)
                return;
        }
    }
}
```

```
if (GoodResult(EZInterface.GetLogEventProperties(Id, ref DeviceType,
    ref DeviceId, ref DeviceNumber, ref DeviceName, ref EventLevel, ref EventType,
    ref EventDesc, ref GeneratedDT, ref ClearedDT, ref ClearedBy, ref AckedBy,
    ref Volume, ref Value, ref ProductVolume, ref ProductLevel, ref WaterLevel,
    ref Temperature)))
{
    WriteMessage("Id: " + Id + ", DType: " + DeviceType + ", DId: " + DeviceId +
        ", DNumber: " + DeviceNumber + ", DName: " + DeviceName +
        ", ELevel: " + EventLevel + ", EType: " + EventType +
        ", EDesc: " + EventDesc + ", GDate: " + GeneratedDT +
        ", CDate: " + ClearedDT + ", CBy: " + ClearedBy +
        ", ABy: " + AckedBy + ", Volume: " + ", Value: " + Value +
        ", PVolume: " + ProductVolume + ", PLevel: " + ProductLevel +
        ", WLevel: " + WaterLevel + ", Temp: " + Temperature);
}
}
```

Lista configurações de combustíveis

Descrição: Esse método apresenta todos os detalhes das configurações de cada combustível configurado no EZServer.

Exemplo:

```
private void ListGrades()
{
    int Idx = 0;
    int Ct = 0;
    int Id = 0;
    int Number = 0;
    Int16 Type = 0;
    String Name = "";
    String ShortName = "";
    String Code = "";
    //-----
    // Ler o numero de produtos configurados //
    if (!GoodResult(EZInterface.GetGradesCount(ref Ct)))
        return;

    WriteMessage("[Produtos " + Ct + "]-----");
    for (Idx = 0; Idx < Ct; Idx++)
    {
        if (EZInterface.GetGradeByOrdinal(Idx + 1, ref Id) != 0)
            return;
    }
}
```

```
if (GoodResult(EZInterface.GetGradePropertiesEx(Id, ref Number, ref Name,  
    ref ShortName, ref Code, ref Type)))  
    WriteMessage(" Grade: " + Number + ", Nome: " + Name + ", Abreviado: " +  
        ShortName + ",Codigo: " + Code + ", Tipo: " + Type);  
}  
WriteMessage("");  
}
```

Lista de sensores

Descrição: Lista de sensores com a quantidade total e suas propriedades.

Exemplo:

```
private void ListSensors()
{
    int Idx = 0;
    int Ct = 0;
    int Id = 0;
    int Number = 0;
    int PortID = 0;
    Int16 Type = 0;
    Int16 Address = 0;
    Int16 SensorNo = 0;
    String Name = "";
    //-----
    // Ler o numero de sensores configurados
    if (!GoodResult(EZInterface.GetSensorsCount(ref Ct)))
        return;

    WriteMessage("[Sensores " + Ct + "]-----");
    for (Idx = 1; Idx <= Ct; Idx++)
    {
        if (EZInterface.GetSensorByOrdinal(Idx, ref Id) != 0)
            return;
        if (GoodResult(EZInterface.GetSensorProperties(Id, ref Number, ref Name,
            ref PortID, ref Type, ref Address, ref SensorNo)))
        {
            WriteMessage(" Sensor: " + Number + ", Nome: " + Name +
                ", Porta: " + PortID + ", Tipo: " + Type +
                ", Endereço: " + Address + ", SensorNo: " + SensorNo);
        }
    }
    WriteMessage("");
}
```

Lista dos ZigBees

Descrição: ZigBee é a rede Wireless da EZTech. Este método pega a lista de todos as redes configuradas no concentrador.

Exemplo:

```
private void ListZigbee()
{
    int Idx = 0;
    int Ct = 0;
    int Id = 0;
    int Number = 0;
    int PortID = 0;
    Int16 DeviceType = 0;
    String Name = "";
    String SerialNumber = "";
    String NodeIdentifier = "";

    //-----
    // Ler o numero de EZRemotes configurados
    if (!GoodResult(EZInterface.GetZigBeeCount(ref Ct)))
        return;

    WriteMessage("[EZRemotes " + Ct + "]-----");
    for (Idx = 0; Idx < Ct; Idx++)
    {
        if (EZInterface.GetZigBeeByOrdinal(Idx, ref Id) != 0)
            return;
        if (GoodResult(EZInterface.GetZigBeeProperties(Id, ref Number, ref Name,
            ref DeviceType, ref SerialNumber, ref NodeIdentifier, ref PortID)))
        {
            WriteMessage(" Zigbee: " + Number + ", Nome: " + Name +
                ", DeviceType: " + DeviceType +
                ", Número de Série: " + SerialNumber +
                ", NodeIdentifier: " + NodeIdentifier + ", PortID: " + PortID);
        }
    }
    WriteMessage("");
}
```

Lista dos bicos

Descrição: Este método pega todas os bicos e suas propriedades configuradas no concentrador.

Exemplo:

```
private void ListHoses()
{
    int Idx = 0;
    int Ct = 0;
    int Id = 0;
    int Number = 0;
    int PumpID = 0;
    int TankID = 0;
    int PhysicalNumber = 0;
    double MtrTheoValue = 0;
    double MtrTheoVolume = 0;
    double MtrElecValue = 0;
    double MtrElecVolume = 0;
    short UVEAntenna = 0;
    double Price1 = 0;
    double Price2 = 0;
    short Enabled = 0;

    //-----
    // Lê o numero de produtos configurados
    if (!GoodResult(EZInterface.GetHosesCount(ref Ct)))
        return;

    WriteMessage("[Bicos = " + Ct + "]" + "-----");
    for (Idx = 1; Idx <= Ct; Idx++)
    {
        if (EZInterface.GetHoseByOrdinal(Id, ref Id) != 0)
            return;
        if (GoodResult(EZInterface.GetHosePropertiesEx2(Id, ref Number, ref PumpID,
            ref TankID, ref PhysicalNumber, ref MtrTheoValue, ref MtrTheoVolume,
            ref MtrElecValue, ref MtrElecVolume, ref UVEAntenna, ref Price1, ref Price2,
            ref Enabled)))
        {
            WriteMessage("  Bico: " + Number + ", PumpID: " + PumpID +
                ", TankID: " + TankID + ", PhysicalNumber: " + PhysicalNumber);
            WriteMessage("    MtrTheoValue: " + MtrTheoValue +
                ", MtrTheoVolume: " + MtrTheoVolume);
            WriteMessage("    MtrElecValue: " + MtrElecValue +
                ", MtrElecVolume: " + MtrElecVolume);
            WriteMessage("    UVEAntena: " + UVEAntenna + ", Price1: " + Price1 +
                ", Price2: " + Price2 + ", Enables: " + Enabled);
        }
    }
}
```

Lista dos encerrantes

Descrição: Encerrante é o total acumulado de produto e venda de uma bomba.

Exemplo:

```
private void btTotals_Click(object sender, EventArgs e)
{
    int IdBomba = 0;
    int IdBico = 0;
    int Bomba = 0;
    int Bico = 0;
    int Number = 0;
    int PumpID = 0;
    int TankID = 0;
    int PhysicalNumber = 0;
    double MtrTheoValue = 0;
    double MtrTheoVolume = 0;
    double MtrElecValue = 0;
    double MtrElecVolume = 0;
    short UVEAntenna = 0;
    double Price1 = 0;
    double Price2 = 0;
    short Enabled = 0;
    Bomba = cbPump.SelectedIndex + 1; // Le o numero da bomba
    // Verifica se esta conectado ao servidor
    if (EZInterface.TestConnection() != 0)
        return;

    // Pega Id da Bomba escolhida
    if (!GoodResult(EZInterface.GetPumpByOrdinal(Bomba, ref IdBomba)))
        return;
    WriteMessage("[ Encerrantes: Bomba " + Bomba +
        " ]-----");
    for (Bico = 1; Bico < 7; Bico++)
    {
        // Le Id do Bico (sem GoodResult() para evitar mensagens no final)
        if (EZInterface.GetPumpHoseByNumber(IdBomba, Bico, ref IdBico)
            != 0)
            return;
        // Le dados do Bico
        if (GoodResult(EZInterface.GetHosePropertiesEx2(IdBico, ref Number,
            ref PumpID, ref TankID, ref PhysicalNumber, ref MtrTheoValue,
            ref MtrTheoVolume, ref MtrElecValue, ref MtrElecVolume,
            ref UVEAntenna, ref Price1, ref Price2, ref Enabled)))
```



```
{
    WriteMessage(" Bico " + Bico + ", EncVolume " + MtrElecVolume
        + ", EncDInheiro " + MtrElecValue + ", Preco1 " + Price1 +
        ", Preco2 " + Price2);
    WriteMessage(" [ Number " + Number + ", PumpId " + PumpID
        + ", TankID " + TankID +
        ", PhysicalNumber " + PhysicalNumber + " ]");
    WriteMessage(" [ MtrTheoValue " + MtrTheoValue +
        ", MtrTheoVolume " + MtrTheoVolume +
        ", UVAntenna " + UVEAntenna + ", Enabled " + Enabled +
        " ]");
    WriteMessage("");
}
}
```

Processamento de eventos

Descrição: Existe uma outra forma de obter as informações do concentrador, usando os eventos você pode pegar os status de um abastecimento, uma bomba entre outras coisas. Abaixo, deixo o primeiro exemplo, nesse caso(conforme o projeto exemplo em C#) é somente a forma de processar os eventos (identificar qual é o evento e direcionar para o método que vai tratá-lo)

Exemplo:

```
private void InternalProcessEvents()
{
    int EvtCt = 0;
    short EvtType = 0;
    // Verifica se esta conectado ao servidor
    if (EZInterface.TestConnection() != 0)
        return;
    // Inicia processamento de eventos
    if (!GoodResult(EZInterface.ProcessEvents()))
        return;
    // Le numero de eventos disponiveis
    if (!GoodResult(EZInterface.GetEventsCount(ref EvtCt)))
        return;

    while (true)
    {
        // Le o proximo evento
        if (!GoodResult(EZInterface.GetNextEventType(ref EvtType)))
            return;
        if (EvtType != 0)
            WriteMessage("-> EVENTO GERADO <- : " + EvtType);
    }
}
```

```
switch ((EZInterface.ClientEvent)EvtType)
{
    //-----
    case EZInterface.ClientEvent.PUMP_EVENT:
        // Trata Eventos das Bombas
        EventPump();
        break;
    //-----
    case EZInterface.ClientEvent.DELIVERY_EVENT:
        // Eventos de abastecimento
        EventDelivery();
        break;
    //-----
    case EZInterface.ClientEvent.CARD_READ_EVENT:
        // Eventos de leitores de cartões
        EventCardRead();
        break;
    //-----
    case EZInterface.ClientEvent.DB_LOG_ETOTALS:
        // Evento de mudança de encerrantes
        EventDbLogETotals();
        break;
    //-----
    case EZInterface.ClientEvent.SERVER_EVENT:
        // Eventos do servidor
        EventServer();
        break;
    //-----
    case EZInterface.ClientEvent.CLIENT_EVENT:
        // Eventos de POS (client)
        EventClient();
        break;
    //-----
    case EZInterface.ClientEvent.ZB2G_STATUS_EVENT:
        // Eventos de Zigbee
        EventZB2G();
        break;
    //-----
    case EZInterface.ClientEvent.LOG_EVENT_EVENT:
        // Log eventos
        EventLog();
        break;
    //-----
    case EZInterface.ClientEvent.DB_TANK_STATUS:
        // Eventos de Tanque
        EventTank();
}
```

```
        break;
//-----
case EZInterface.ClientEvent.NO_CLIENT_EVENT:
    // Trata Eventos do Cliente
    return;
//-----
default:
    GoodResult(EZInterface.DiscardNextEvent());
    WriteMessage("Não há eventos");
    break;
    }
    }
}
```

Eventos da bomba

Descrição: Eventos gerados pela bomba, o mesmo serve para pegar todas as informações da bomba.

Exemplo:

```
private void EventPump()
{
    int PumpID = 0;
    int PumpNumber = 0;
    short State = 0;
    short ReservedFor = 0;
    int ReservedBy = 0;
    int HoseID = 0;
    int HoseNumber = 0;
    int HosePhysicalNumber = 0;
    int GradeID = 0;
    int GradeNumber = 0;
    short PriceLevel = 0;
    double Price = 0;
    double Volume = 0;
    double Value = 0;
    short StackSize = 0;
    int PhysicalNumber = 0;
    short Side = 0;
    short Address = 0;
    short PriceLevel1 = 0;
    short PriceLevel2 = 0;
    short PumpType = 0;
    int PortID = 0;
    short AuthMode = 0;
    short StackMode = 0;
```

```
short PrepayAllowed = 0;
short PreauthAllowed = 0;
short PriceFormat = 0;
short ValueFormat = 0;
short VolumeFormat = 0;
Int64 Tag = 0;
int AttendantID = 0;
int AttendantNumber = 0;
Int64 AttendantTag = 0;
int CardClientID = 0;
int CardClientNumber = 0;
Int64 CardClientTag = 0;
double CurFlowRate = 0;
double PeakFlowRate = 0;
String GradeName = "";
String ShortGradeName = "";
String PumpName = "";
String AttendantName = "";
String CardClientName = "";
// Verifica se esta conectado ao servidor
if (EZInterface.TestConnection() != 0)
    return;
if (GoodResult(EZInterface.GetNextPumpEventEx3(ref PumpID, ref PumpNumber,
    ref State, ref ReservedFor, ref ReservedBy, ref HoseID, ref HoseNumber,
    ref HosePhysicalNumber, ref GradeID, ref GradeNumber, ref GradeName,
    ref ShortGradeName, ref PriceLevel, ref Price, ref Volume, ref Value,
    ref StackSize, ref PumpName, ref PhysicalNumber, ref Side, ref Address,
    ref PriceLevel1, ref PriceLevel2, ref PumpType, ref PortID, ref AuthMode,
    ref StackMode, ref PrepayAllowed, ref PreauthAllowed, ref PriceFormat,
    ref ValueFormat, ref VolumeFormat, ref Tag, ref AttendantID,
    ref AttendantNumber, ref AttendantName, ref AttendantTag, ref CardClientID,
    ref CardClientNumber, ref CardClientName, ref CardClientTag,
    ref CurFlowRate, ref PeakFlowRate)))
{
    WriteMessage("    PumpEvent: " +
        " PumpID= " + PumpID +
        ", PumpNumber= " + PumpNumber +
        ", State= " + State +
        ", ReservedFor= " + ReservedFor +
        ", ReservedBy= " + ReservedBy +
        ", HoseID= " + HoseID +
        ", HoseNumber= " + HoseNumber +
        ", HosePhysicalNumber= " + HosePhysicalNumber +
        ", GradeID= " + GradeID +
        ", GradeName= " + GradeName +
```

```
    ", GradeNumber= " + GradeNumber +  
    ", ShortGradeName= " + ShortGradeName +  
    ", PriceLevel= " + PriceLevel +  
    ", Price= " + Price +  
    ", Volume= " + Volume +  
    ", Value= " + Value +  
    ", StackSize= " + StackSize +  
    ", PumpName= " + PumpName +  
    ", PhysicalNumber= " + PhysicalNumber +  
    ", Side= " + Side +  
    ", Address= " + Address +  
    ", PriceLevel1= " + PriceLevel1 +  
    ", PriceLevel2= " + PriceLevel2 +  
    ", PumpType= " + PumpType +  
    ", PortID= " + PortID +  
    ", AuthMode= " + AuthMode +  
    ", StackMode= " + StackMode +  
    ", PrepayAllowed= " + PrepayAllowed +  
    ", PreauthAllowed= " + PreauthAllowed +  
    ", PriceFormat= " + PriceFormat +  
    ", ValueFormat= " + ValueFormat +  
    ", VolumeFormat= " + VolumeFormat +  
    ", Tag= " + Tag +  
    ", AttendantID= " + AttendantID +  
    ", AttendantNumber= " + AttendantNumber +  
    ", AttendantName= " + AttendantName +  
    ", AttendantTag= " + AttendantTag +  
    ", CardClientID= " + CardClientID +  
    ", CardClientNumber= " + CardClientNumber +  
    ", CardClientName= " + CardClientName +  
    ", CardClientTag= " + CardClientTag +  
    ", CurFlowRate= " + CurFlowRate +  
    ", PeakFlowRate= " + PeakFlowRate);
```

```
    WriteMessage("      Bico Equivalente CBC: " +  
CompanyID((short)HoseNumber, (short)PumpNumber));  
  }  
}
```

Eventos da abastecimento

Descrição: Eventos gerados pelo abastecimento, o mesmo serve para pegar todas as informações voltadas ao abastecimento.

Exemplo:

```
private void EventDelivery()  
{  
    int DeliveryID = 0;  
    int HosePhysicalNumber = 0;  
    int TankID = 0;  
    int TankNumber = 0;  
    short DeliveryState = 0;  
    short DeliveryType = 0;  
    double Volume2 = 0;  
    DateTime CompletedDT = new DateTime();  
    int LockedBy = 0;  
    int Age = 0;  
    DateTime ClearedDT = new DateTime();  
    double OldVolumeETot = 0;  
    double OldVolume2ETot = 0;  
    double OldValueETot = 0;  
    double NewVolumeETot = 0;  
    double NewVolume2ETot = 0;  
    double NewValueETot = 0;  
    int Duration = 0;  
    int PumpID = 0;  
    int PumpNumber = 0;  
    int HoseID = 0;  
    int HoseNumber = 0;  
    int GradeID = 0;  
    int GradeNumber = 0;  
    short PriceLevel = 0;  
    double Price = 0;  
    double Volume = 0;  
    double Value = 0;  
    int ReservedBy = 0;  
    Int64 Tag = 0;  
    int AttendantID = 0;  
    int AttendantNumber = 0;  
    Int64 AttendantTag = 0;  
    int CardClientID = 0;  
    int CardClientNumber = 0;  
    Int64 CardClientTag = 0;  
    double PeakFlowRate = 0;  
    short State = 0;  
}
```

```
short Type = 0;
String TankName = "";
String GradeShortName = "";
String GradeCode = "";
String PumpName = "";
String GradeName = "";
String AttendantName = "";
String CardClientName = "";

// Verifica se esta conectado ao servidor
if (EZInterface.TestConnection() != 0)
    return;

if (GoodResult(EZInterface.GetNextDeliveryEventEx3(ref DeliveryID, ref HoseID,
    ref HoseNumber, ref HosePhysicalNumber, ref PumpID, ref PumpNumber,
    ref PumpName, ref TankID, ref TankNumber, ref TankName,
    ref GradeID, ref GradeNumber, ref GradeName, ref GradeShortName,
    ref GradeCode, ref State, ref Type, ref Volume, ref PriceLevel,
    ref Price, ref Value, ref Volume2, ref CompletedDT, ref LockedBy,
    ref ReservedBy, ref AttendantID, ref Age, ref ClearedDT,
    ref OldVolumeETot, ref OldVolume2ETot, ref OldValueETot,
    ref NewVolumeETot, ref NewVolume2ETot, ref NewValueETot,
    ref Tag, ref Duration, ref AttendantNumber, ref AttendantName,
    ref AttendantTag, ref CardClientID, ref CardClientNumber,
    ref CardClientName, ref CardClientTag))))
{
    // Primeiro abastecimento pode ser invalido
    if (DeliveryID > 0)
    {
        WriteMessage("    DeliveryEvent: " +
            " DeliveryID= " + DeliveryID +
            ", HoseID= " + HoseID +
            ", HoseNumber= " + HoseNumber +
            ", HosePhysicalNumber= " + HosePhysicalNumber +
            ", PumpID= " + PumpID +
            ", PumpNumber= " + PumpNumber +
            ", PumpName= " + PumpName +
            ", TankID= " + TankID +
            ", TankNumber= " + TankNumber +
            ", TankName= " + TankName +
            ", GradeID= " + GradeID +
            ", GradeNumber= " + GradeNumber +
            ", GradeName= " + GradeName +
            ", GradeShortName= " + GradeShortName +
            ", GradeCode= " + GradeCode +
            ", DeliveryState= " + DeliveryState +
```

```
        ", DeliveryType= " + DeliveryType +  
        ", Volume= " + Volume +  
        ", PriceLevel= " + PriceLevel +  
        ", Price= " + Price +  
        ", Value= " + Value +  
        ", Volume2= " + Volume2 +  
        ", CompletedDT= " + CompletedDT +  
        ", LockedBy= " + LockedBy +  
        ", ReservedBy= " + ReservedBy +  
        ", AttendantID= " + AttendantID +  
        ", Age= " + Age +  
        ", ClearedDT= " + ClearedDT +  
        ", OldVolumeETot= " + OldVolumeETot +  
        ", OldVolume2ETot= " + OldVolume2ETot +  
        ", OldValueETot= " + OldValueETot +  
        ", NewVolumeETot= " + NewVolumeETot +  
        ", NewVolume2ETot= " + NewVolume2ETot +  
        ", NewValueETot= " + NewValueETot +  
        ", Tag= " + Tag +  
        ", Duration= " + Duration +  
        ", AttendantNumber= " + AttendantNumber +  
        ", AttendantName= " + AttendantName +  
        ", AttendantTag= " + AttendantTag +  
        ", CardClientID= " + CardClientID +  
        ", CardClientNumber= " + CardClientNumber +  
        ", CardClientName= " + CardClientName +  
        ", CardClientTag= " + CardClientTag +  
        ",PeakFlowRate= " + PeakFlowRate);  
  
    WriteMessage("        Bico Equivalente CBC: " +  
CompanyID((short)HoseNumber, (short)PumpNumber));  
  
    }  
}  
}
```


Eventos de leitura dos cartões

Descrição: Eventos gerados para a leitura dos cartões, sempre que um cartão for lido ele irá gerar um evento que entrará nesta condição.

Exemplo:

```
private void EventCardRead()
{
    int CardReadID = 0;
    int Number = 0;
    short CardType = 0;
    int ParentID = 0;
    DateTime TimeStamp = new DateTime();
    Int64 Tag = 0;
    int PumpID = 0;
    String Name = "";
    // Verifica se esta conectado ao servidor
    if (EZInterface.TestConnection() != 0)
        return;

    WriteMessage("Leitura de cartão solicitada!");

    if (GoodResult(EZInterface.GetNextCardReadEvent(ref CardReadID, ref Number,
    ref Name, ref PumpID, ref CardType, ref ParentID, ref Tag, ref TimeStamp)))
    {
        WriteMessage("\n----- CardReadEvent: CardReadID " + CardReadID +
            ", Number " + Number +
            ", Name " + Name +
            ", PumpID " + PumpID);
        WriteMessage("CardType " + CardType +
            ", ParentID " + ParentID +
            ", Tag " + Tag +
            ", TimeStamp " + TimeStamp);

        // <summary>
        // Rotina utilizada para identificar o status do cartão lido.
        // </summary>
        switch ((EZInterface.TTagType)CardType)
        {
            case EZInterface.TTagType.ATTENDANT_TAG_TYPE:
                WriteMessage("Attendant: " + Name + " Tag " + Tag);
                break;
            case EZInterface.TTagType.BLOCKED_ATTENDANT_TAG_TYPE:
                WriteMessage("\nBlocked attendant: " + Name + " Tag " + Tag);
                break;
```

```
case EZInterface.TTagType.WRONG_SHIFT_ATTENDANT_TAG_TYPE:
    WriteMessage("\n      Wrong shift attendant: " + Name + " Tag " + Tag);
    break;
case EZInterface.TTagType.CLIENT_TAG_TYPE:
    WriteMessage("\n      Client: " + Name + " Tag " + Tag);
    break;
case EZInterface.TTagType.BLOCKED_CLIENT_TAG_TYPE:
    WriteMessage("\n      Blocked Client: " + Name + " Tag " + Tag);
    break;
case EZInterface.TTagType.UNKNOWN_TAG_TYPE:
    WriteMessage("\n      Unknown Tag read: " + Tag);
    break;
default:
    WriteMessage("\n      Unknown Tag type: " + CardType + " Tag " + Tag);
    break;
}
GoodResult(EZInterface.DeleteCardRead(CardReadID));
}
}
```

Eventos de mudança de encerrantes

Descrição: Método para captura dos eventos de mudança dos encerrantes.

Exemplo:

```
private void EventDbLogETotals()
{
    int HoseID = 0;
    double Volume = 0;
    double Value = 0;
    double VolumeETot = 0;
    double ValueETot = 0;
    int HoseNumber = 0;
    int HosePhysicalNumber = 0;
    int PumpID = 0;
    int PumpNumber = 0;
    int TankID = 0;
    int TankNumber = 0;
    int GradeID = 0;
    String PumpName = "";
    String TankName = "";
    String GradeName = "";
    // Verifica se esta conectado ao servidor
    if (EZInterface.TestConnection() != 0)
        return;
```

```
if (GoodResult(EZInterface.GetNextDBHoseETotalsEventEx(ref HoseID, ref Volume,
    ref Value, ref VolumeETot, ref ValueETot, ref HoseNumber,
    ref HosePhysicalNumber, ref PumpID, ref PumpNumber, ref PumpName,
    ref TankID, ref TankNumber, ref TankName, ref GradeID, ref GradeName)))
{
    WriteMessage("----- HoseETotalEvent: HoseID " + HoseID + ", Volume " +
Volume + " Value " + Value);
    WriteMessage("          VolumeETot " + VolumeETot + ", ValueETot " +
ValueETot);
    WriteMessage("          HoseNumber " + HoseNumber + ", HosePhysicalNumber
" + HosePhysicalNumber);
    WriteMessage("          PumpID " + PumpID + ", PumpNumber " + PumpNumber
+ ", PumpName " + PumpName);
    WriteMessage("          TankID " + TankID + ", TankNumber " + TankNumber + ",
TankName " + TankName);
    WriteMessage("          GradeID " + GradeID + ", GradeName " + GradeName);
}
}
```

Eventos do EZServer

Descrição: Método para captura dos eventos gerados pelo EZServer.

Exemplo:

```
private void EventServer()
{
    int EventID = 0;
    String EventText= "";
    // Verifica se esta conectado ao servidor
    if (EZInterface.TestConnection() != 0)
        return;
    if (GoodResult(EZInterface.GetNextServerEvent(ref EventID, ref EventText)))
        WriteMessage("----- ServerEvent: EventID " + EventID + ", EventText " +
EventText);
}
```

Eventos do Client

Descrição: Este evento é gerado quando outro cliente EZServer chama FireClientEvent. O valor do EventID e EventText são simplesmente passados.

Exemplo:

```
private void EventClient()
{
    int EventID = 0;
    short ClientID = 0;
    String EventText = "";
    // Verifica se esta conectado ao servidor
    if (EZInterface.TestConnection() != 0)
        return;

    if (GoodResult(EZInterface.GetNextClientEvent(ref ClientID, ref EventID,
    ref EventText)))
        WriteMessage("----- ClientEvent: ClientID " + ClientID + ", EventID " + EventID +
        ", EventText " + EventText);
}
```

Eventos dos dispositivos ZigBee

Descrição: Este evento é acionado quando o status de um dispositivo ZigBee específico é alterado.

Exemplo:

```
private void EventZB2G()
{
    Int32 PortID = 0;
    Int64 ZBAddress = 0;
    Int16 LQI = 0;
    Int16 RSSI = 0;
    Int64 ParZBAddress = 0;
    Int16 ZBChannel = 0;
    Int16 MemBlocks = 0;
    Int16 MemFree = 0;
    // Verifica se esta conectado ao servidor
    if (EZInterface.TestConnection() != 0)
        return;

    if (GoodResult(EZInterface.GetNextZB2GStatusEvent(ref PortID, ref ZBAddress,
    ref LQI, ref RSSI, ref ParZBAddress, ref ZBChannel, ref MemBlocks, ref MemFree)))
        WriteMessage("----- ZigBeeEvent: PortID " + PortID + ", Endereço ZigBee " +
        ZBAddress + ", LQI " + LQI + ", RSSI " + RSSI + ", ParZBAddress " + ParZBAddress
        + ", Canal " + ZBChannel + ", Memória Bloqueada " + MemBlocks + ", Memória Livre
        " + MemFree);
}
```

Eventos de Log

Descrição: Este evento é acionado sempre que um evento de log é criado ou alterado.

Exemplo:

```
private void EventLog()
{
    Int32 LogEventID = 0;
    Int16 DeviceType = 0;
    Int32 DeviceID = 0;
    Int32 DeviceNumber = 0;
    String DeviceName = "";
    Int16 EventLevel = 0;
    Int16 EventType = 0;
    String EventDesc = "";
    DateTime GeneratedDT = new DateTime();
    DateTime ClearedDT = new DateTime();
    Int32 ClearedBy = 0;
    Int32 AckedBy = 0;
    Double Volume = 0;
    Double Value = 0;
    Double ProductVolume = 0;
    Double ProductLevel = 0;
    Double WaterLevel = 0;
    Double Temperature = 0;
    // Verifica se esta conectado ao servidor
    if (EZInterface.TestConnection() != 0)
        return;
    if (GoodResult(EZInterface.GetNextLogEventEvent(ref LogEventID,
        ref DeviceType, ref DeviceID, ref DeviceNumber, ref DeviceName,
        ref EventLevel, ref EventType, ref EventDesc, ref GeneratedDT,
        ref ClearedDT, ref ClearedBy, ref AckedBy, ref Volume, ref Value,
        ref ProductVolume, ref ProductLevel, ref WaterLevel, ref Temperature)))
    {
        WriteMessage("----- LogEvent: LogEventID " + LogEventID +
            ", DeviceType " + DeviceType + ", DeviceID " + DeviceID +
            ", DeviceNumber " + DeviceNumber + ", DeviceName " + DeviceName +
            ", EventLevel " + EventLevel + ", EventType " + EventType +
            ", EventDesc " + EventDesc + ", GeneratedDT " + GeneratedDT +
            ", ClearedDT " + ClearedDT + ", ClearedBy " + ClearedBy +
            ", AckedBy " + AckedBy + ", Volume " + Volume +
            ", Value " + Value + ", ProductVolume " + ProductVolume +
            ", ProductLevel " + ProductLevel + ", WaterLevel " + WaterLevel +
            ", Temperature " + Temperature);
    }
}
```