

TRILOBIT

Manual do SDK para REP Trilobit
Módulo: API RepTrilobit

Março/2010

TRILOBIT

Conteúdo

1. Apresentação	2
2. Propriedades	3
2.1. Exception	3
3. Métodos de envio	4
3.1. Cadastro de empregador	4
3.2. Cadastro de empregado	5
3.3. Exclusão de empregado	6
3.4. Envio de configuração	7
4. Métodos de leitura	8
4.1. Leitura de AFD	8
4.1.1. Ler AFD via arquivo	8
4.1.2. Ler AFD via lista	9
4.1.3. Ler AFD via DataTable (Exclusivo para .NET)	10
4.2. Leitura de empregados	11
4.2.1. Ler empregados via arquivo	11
4.2.2. Ler empregados via lista	12
4.2.3. Ler empregados via DataTable (Exclusivo para .NET)	13
4.2.4. Ler empregado específico pelo crachá	14
4.3. Leitura de configurações	15
4.3.1. Ler configurações via arquivo	15
4.3.2. Ler configurações via lista	16
4.3.3. Ler configurações via DataTable (Exclusivo para .NET)	17
4.3.4. Ler configuração específica	18
5. Tipos de dados	19
5.1. Inteiro	19
5.2. Booleano	19
5.3. Texto	19
5.4. DataTable (Exclusivo para .NET)	19
6. Enumeradores	20
6.1. eTipoDocumento	20
6.2. eParamSetConfig	21
6.3. eParamGetConfig	22
7. Formas de Utilização	24
7.1. C# (Utilizando Visual Studio 2008)	24
7.2. Visual Basic 6	26
7.3. Delphi (Utilizando Delphi 2007)	29

1. Apresentação

Este manual tem por objetivo detalhar o uso da API (interface de programação) RepTrilobit, desenvolvida para comunicação com o REP (Registrador Eletrônico de Ponto) da Trilobit.

A maioria dos métodos da API RepTrilobit pode ser utilizada tanto em plataforma .NET como em plataforma COM. Os métodos que não são compatíveis com a plataforma COM possuem uma observação, indicando que são exclusivos para .NET.

Obs.: Para maiores informações sobre a arquitetura do SDK para REP Trilobit, consulte o documento “GuiaReferencia_SDK_RepTrilobit.pdf”

2. Propriedades

Neste tópico veremos quais são e como utilizar as propriedades da API RepTrilobit.

2.1. `ErrorException`

Retorna o último erro ocorrido dentro da API RepTrilobit.

Sintaxe

`object.ErrorException`

Onde

`object` - Instância da API RepTrilobit.

Retorno

Um objeto do tipo `Exception`.

Formas de uso

`object.ErrorException.Message;`

3. Métodos de envio

Neste tópico veremos quais são e como utilizar os métodos de envio de informações da API RepTrilobit.

3.1. Cadastro de empregador

Sintaxe

object.CadastrarEmpregador(IP, Porta, Senha, TipoDoc, Documento, CEI, RazaoSocial, Local)

Descrição

Envia os dados do empregador para o REP indicado.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

TipoDoc - Um Enumerador (eTipoDocumento) que indica o tipo de documento do empregador.

Documento - Um Texto, com no máximo 14 dígitos, que indica o número do documento do empregador. O número do documento pode estar formatado como CNPJ (99.999.999/9999-99) / CPF (999.999.999-99) ou apenas os números, sem traços, pontos ou barras.

CEI - Um Texto, com no máximo 12 dígitos, que indica o CEI do empregador, sem traços, pontos ou barras, apenas números.

RazaoSocial - Um Texto, com no máximo 150 caracteres, que indica a razão social do empregador.

Local - Um Texto, com no máximo 100 caracteres, que indica o local de prestação de serviço do empregador.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.CadastrarEmpregador(IP, Porta, Senha, TipoDoc, Documento, CEI, RazaoSocial, Local)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

Para maiores detalhes sobre os valores possíveis para o parâmetro *TipoDoc*, consulte o tópico “Enumeradores → eTipoDocumento”.

3.2. Cadastro de empregado

Sintaxe

object.CadastrarEmpregado(IP, Porta, Senha, PIS, Nome, Cracha, PossuiBiometria)

Descrição

Envia os dados de um empregado para o REP indicado.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

PIS - Um Texto, com no máximo 12 dígitos, que indica o número do documento do empregado, sem traços, pontos ou barras, apenas números.

Nome - Um Texto, com no máximo 52 caracteres, que indica o nome do empregado.

Cracha - Um Texto, com no máximo 20 caracteres, que indica o número do crachá de identificação do empregado.

PossuiBiometria - Um Booleano que indica se o empregado possui ou não biometria.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.CadastrarEmpregado(IP, Porta, Senha, PIS, Nome, Cracha, PossuiBiometria)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

3.3. Exclusão de empregado

Sintaxe

object.ExcluirEmpregado(IP, Porta, Senha, PIS)

Descrição

Exclui os dados de um empregado do REP indicado.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

PIS - Um Texto, com no máximo 12 dígitos, que indica o número do documento do empregado, sem traços, pontos ou barras, apenas números.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.ExcluirEmpregado(IP, Porta, Senha, PIS)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterà um objeto com o erro ocorrido.

3.4. Envio de configuração

Sintaxe

object.EnviarConfiguracao(*IP*, *Porta*, *Senha*, *Parametro*, *Valor*)

Descrição

Envia um parâmetro de configuração ao REP indicado.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Parametro - Um Enumerador(eParamSetConfig) que indica qual parâmetro está sendo enviado.

Valor - Um Texto que indica qual valor está sendo enviado.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object*.EnviarConfiguracao(*IP*, *Porta*, *Senha*, *Parametro*, *Valor*)

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

Para maiores detalhes sobre os valores possíveis para o parâmetro *Parametro*, consulte o tópico “Enumeradores → eParamSetConfig”.

4. Métodos de leitura

Neste tópico veremos quais são e como utilizar os métodos de leitura de informações da API RepTrilobit.

4.1. Leitura de AFD

Neste tópico veremos quais são e como utilizar os métodos de leitura de AFD da API RepTrilobit.

4.1.1. Ler AFD via arquivo

Sintaxe

object.LerAFD(IP, Porta, Senha, DataInicial, DataFinal, ArquivoAFD)

Descrição

Gera um arquivo texto, formatado conforme Anexo I da Portaria nº1.510, de 21 de Agosto de 2009, contendo o AFD do REP indicado, nas datas informadas.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

DataInicial - Um Texto que indica a data inicial para geração do arquivo. A data deve estar no formato AAAAMMDD.

DataFinal - Um Texto que indica a data final para geração do arquivo. A data deve estar no formato AAAAMMDD.

ArquivoAFD - Um Texto que indica o caminho e nome do arquivo que deve ser gerado.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = object.LerAFD(IP, Porta, Senha, DataInicial, DataFinal, ArquivoAFD)

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.1.2. Ler AFD via lista

Sintaxe

object.LerAFD_ViaLista(IP, Porta, Senha, DataInicial, DataFinal, ref Lista, NSR, SeparadorCampo, SeparadorRegistro)

Descrição

Gera um Texto, contendo os dados de AFD do REP indicado, nas datas informadas e com indicação de um NSR para início de pesquisa, separados pelos caracteres especificados.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

DataInicial - Um Texto que indica a data inicial para geração do arquivo. A data deve estar no formato AAAAMMDD.

DataFinal - Um Texto que indica a data final para geração do arquivo. A data deve estar no formato AAAAMMDD.

Lista - Um Texto, passado por referência, que retorna as informações geradas.

NSR - Um Inteiro que indica o último NSR lido. O AFD será gerado a partir do próximo NSR existente. Indicar "0" (zero) para a primeira leitura.

SeparadorCampo - Um Texto que indica qual caracter deve ser utilizado para separar os campos da lista. Caso nenhum caracter seja informado, será utilizado o caracter ";" (ponto e vírgula).

SeparadorRegistro - Um Texto que indica qual caracter deve ser utilizado para separar os registros da lista. Caso nenhum caracter seja informado, será utilizado o caracter "|" (pipe).

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = object.LerAFD_ViaLista(IP, Porta, Senha, DataInicial, DataFinal, ref Lista, NSR, SeparadorCampo, SeparadorRegistro)

Observações

Caso o retorno seja **False**, a propriedade **ErrorException** conterá um objeto com o erro ocorrido.

4.1.3. Ler AFD via DataTable (Exclusivo para .NET)

Sintaxe

object.LerAFD(IP, Porta, Senha, DataInicial, DataFinal, ref Tabela, NSR)

Descrição

Gera um DataTable, com estrutura de 11 campos do tipo Texto, contendo o AFD do REP indicado, nas datas informadas e com indicação de um NSR para início de pesquisa.

A estrutura de cada registro do DataTable varia conforme o tipo de registro, conforme Anexo I da Portaria nº 1.510, de 21 de Agosto de 2009.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

DataInicial - Um Texto que indica a data inicial para geração do arquivo. A data deve estar no formato AAAAMMDD.

DataFinal - Um Texto que indica a data final para geração do arquivo. A data deve estar no formato AAAAMMDD.

Tabela - Um DataTable, passado por referência, que será preenchido com os dados do AFD.

NSR - Um Inteiro que indica o último NSR lido. O AFD será gerado a partir do próximo NSR existente. Indicar "0" (zero) para a primeira leitura.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = object.LerAFD(IP, Porta, Senha, DataInicial, DataFinal, ref Tabela, NSR)

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.2. Leitura de empregados

Neste tópico veremos quais são e como utilizar os métodos de leitura de empregados da API RepTrilobit.

4.2.1. Ler empregados via arquivo

Sintaxe

object.LerEmpregados(IP, Porta, Senha, Arquivo, AdicionarCabecalho)

Descrição

Gera um arquivo texto, formatado conforme layout definido no tópico “Exportando Arquivo Empregados” do documento “Manual_ReplIntegracao.pdf”, contendo todos os empregados cadastrados no REP indicado.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Arquivo - Um Texto que indica o caminho e nome do arquivo que deve ser gerado.

AdicionarCabecalho - Um Booleano que indica se o arquivo gerado deverá conter o cabeçalho utilizado pelo módulo “ReplIntegracao” (ReplIntegracao.exe).

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.LerEmpregados(IP, Porta, Senha, Arquivo, AdicionarCabecalho)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.2.2. Ler empregados via lista

Sintaxe

object.LerEmpregados_ViaLista(IP, Porta, Senha, ref Lista, SeparadorCampo, SeparadorRegistro)

Descrição

Gera um Texto, contendo os dados de todos os empregados cadastrados no REP indicado, separados pelos caracteres especificados.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Lista - Um Texto, passado por referência, que retorna as informações geradas.

SeparadorCampo - Um Texto que indica qual caracter deve ser utilizado para separar os campos da lista. Caso nenhum caracter seja informado, será utilizado o caracter “;” (ponto e vírgula).

SeparadorRegistro - Um Texto que indica qual caracter deve ser utilizado para separar os registros da lista. Caso nenhum caracter seja informado, será utilizado o caracter “|” (pipe).

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = object.LerEmpregados_ViaLista(IP, Porta, Senha, ref Lista, SeparadorCampo, SeparadorRegistro)

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.2.3. Ler empregados via DataTable (Exclusivo para .NET)

Sintaxe

object.LerEmpregados(IP, Porta, Senha, ref Tabela)

Descrição

Gera um DataTable, com estrutura de 4 campos do tipo Texto, contendo todos os empregados cadastrados no REP indicado.

Os campos definidos no DataTable são: PIS, Nome, Cracha e PossuiBiometria.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Tabela - Um DataTable, passado por referência, que será preenchido com os dados dos empregados.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.LerEmpregados(IP, Porta, Senha, ref Tabela)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.2.4. Ler empregado específico pelo crachá

Sintaxe

object.LerEmpregado_PeloCracha(IP, Porta, Senha, Cracha, ref PIS)

Descrição

Retorna o PIS do empregado que estiver utilizando o crachá informado. Caso nenhum empregado esteja utilizando o crachá informado, o parâmetro PIS é retornado em branco.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Cracha - Um Texto, com no máximo 20 caracteres, que indica o crachá de identificação que se deseja verificar.

PIS - Um Texto, passado por referência, que será preenchido com o PIS do empregado que estiver associado ao crachá informado.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.LerEmpregado_PeloCracha(IP, Porta, Senha, Cracha, ref PIS)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.3. Leitura de configurações

Neste tópico veremos quais são e como utilizar os métodos de leitura de configurações da API RepTrilobit.

4.3.1. Ler configurações via arquivo

Sintaxe

object.LerConfiguracoes(IP, Porta, Senha, Arquivo, AdicionarCabecalho)

Descrição

Gera um arquivo texto, formatado conforme layout definido no tópico “Exportando Arquivo Configurações” do documento “Manual_ReplIntegracao.pdf”, contendo todas as configurações do REP indicado.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Arquivo - Um Texto que indica o caminho e nome do arquivo que deve ser gerado.

AdicionarCabecalho - Um Booleano que indica se o arquivo gerado deverá conter o cabeçalho utilizado pelo módulo “ReplIntegracao” (ReplIntegracao.exe).

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.LerConfiguracoes(IP, Porta, Senha, Arquivo, AdicionarCabecalho)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.3.2. Ler configurações via lista

Sintaxe

object.LerConfiguracoes_ViaLista(IP, Porta, Senha, ref Lista, SeparadorCampo, SeparadorRegistro)

Descrição

Gera um Texto contendo os dados de todas as configurações do REP indicado, separados pelos caracteres especificados.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Lista - Um Texto, passado por referência, que retorna as informações geradas.

SeparadorCampo - Um Texto que indica qual caracter deve ser utilizado para separar os campos da lista. Caso nenhum caracter seja informado, será utilizado o caracter “;” (ponto e vírgula).

SeparadorRegistro - Um Texto que indica qual caracter deve ser utilizado para separar os registros da lista. Caso nenhum caracter seja informado, será utilizado o caracter “|” (pipe).

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = object.LerConfiguracoes_ViaLista(IP, Porta, Senha, ref Lista, SeparadorCampo, SeparadorRegistro)

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.3.3. Ler configurações via DataTable (Exclusivo para .NET)

Sintaxe

object.LerConfiguracoes(IP, Porta, Senha, ref Tabela)

Descrição

Gera um DataTable, com estrutura de 2 campos do tipo Texto, contendo todas as configurações do REP indicado.

Os campos definidos no DataTable são: Parametro, Valor.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Tabela - Um DataTable, passado por referência, que será preenchido com os dados das configurações.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.LerConfiguracoes(IP, Porta, Senha, ref Tabela)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

4.3.4. Ler configuração específica

Sintaxe

object.LerConfiguracao(IP, Porta, Senha, Parametro, ref Valor)

Descrição

Lê um parâmetro de configuração do REP indicado.

Onde

object - Instância da API RepTrilobit.

IP - Um Texto que indica o endereço IP do REP.

Porta - Um Inteiro que indica a porta UDP do REP.

Senha - Um Inteiro que indica a senha do REP.

Parametro - Um Enumerador (eParamGetConfig) que indica qual parâmetro está sendo lido.

Valor - Um Texto, passado por referência, que retorna qual valor está configurado no REP.

Retorno

Um Booleano indicando se o comando foi enviado com sucesso para o REP.

Formas de uso

retorno = *object.LerConfiguracao(IP, Porta, Senha, Parametro, ref Valor)*

Observações

Caso o retorno seja **False**, a propriedade `ErrorException` conterá um objeto com o erro ocorrido.

Para maiores detalhes sobre os valores possíveis para o parâmetro *Parametro*, consulte o tópico “Enumeradores → eParamGetConfig”.

5. Tipos de dados

Neste tópico veremos quais são os tipos de dados utilizados pela API RepTrilobit.

5.1. Inteiro

Valor inteiro de 32 bits (4 bytes), variável entre -2.147.483.648 e 2.147.483.647

Linguagem	Observações
C#	Int32 ou int.
Basic	Long.
Pascal	Integer.

5.2. Booleano

Valor lógico, podendo ser True (Verdadeiro) ou False (Falso).

Linguagem	Observações
C#	Boolean ou bool.
Basic	Boolean.
Pascal	WordBool.

5.3. Texto

Cadeia variável de caracteres, podendo conter até aproximadamente 2 bilhões de caracteres (2³¹) ASCII.

Linguagem	Observações
C#	String ou string.
Basic	String.
Pascal	WideString.

5.4. DataTable (Exclusivo para .NET)

Tabela de dados em memória, pertencente ao namespace System.Data da .NET Framework.

Linguagem	Observações
C#	DataTable.

6. Enumeradores

Neste tópico veremos quais são e quais valores possuem os enumeradores da API RepTrilobit.

6.1. eTipoDocumento

Enumerador utilizado para indicar os valores possíveis para o tipo de documento do empregador.

Tipo de dado

Inteiro.

Valores possíveis

1 - CNPJ

2 - CPF

Observações

Em Delphi, ao invés do enumerador, existem constantes que representam os valores acima. As constantes possuem o mesmo nome do enumerador, precedidas de “eTipoDocumento_”.

Por exemplo:

eTipoDocumento_CNPJ

6.2. eParamSetConfig

Enumerador utilizado para indicar os valores possíveis para o envio de configurações para o REP.

Tipo de dado

Inteiro.

Valores possíveis

Código	Valor	Tamanho	Descrição																																
EnderecolP	1	15	Endereço IP do REP. Formato: 000.000.000.000																																
PortaUDP	2	5	Porta UDP do REP (De 1 a 65536). Exemplo: 19001																																
MascaraRede	3	15	Máscara de rede do REP. Formato: 000.000.000.000																																
Roteador	4	15	Endereço do roteador do REP. Formato: 000.000.000.000																																
Tipoidentificacao	5	1	Tipo de identificação aceita pelo REP. Valores possíveis: <table border="1"> <tr> <th>Valor</th><th>Teclado</th><th>Leitor</th><th>Bio</th></tr> <tr> <td>1</td><td>OK</td><td>NOK</td><td>NOK</td></tr> <tr> <td>2</td><td>NOK</td><td>OK</td><td>NOK</td></tr> <tr> <td>3</td><td>OK</td><td>OK</td><td>NOK</td></tr> <tr> <td>4</td><td>NOK</td><td>NOK</td><td>OK</td></tr> <tr> <td>5</td><td>OK</td><td>NOK</td><td>OK</td></tr> <tr> <td>6</td><td>NOK</td><td>OK</td><td>OK</td></tr> <tr> <td>7</td><td>OK</td><td>OK</td><td>OK</td></tr> </table>	Valor	Teclado	Leitor	Bio	1	OK	NOK	NOK	2	NOK	OK	NOK	3	OK	OK	NOK	4	NOK	NOK	OK	5	OK	NOK	OK	6	NOK	OK	OK	7	OK	OK	OK
Valor	Teclado	Leitor	Bio																																
1	OK	NOK	NOK																																
2	NOK	OK	NOK																																
3	OK	OK	NOK																																
4	NOK	NOK	OK																																
5	OK	NOK	OK																																
6	NOK	OK	OK																																
7	OK	OK	OK																																
InicioHorarioVerao	6	14	Data e hora do início do horário de verão. Formato: AAAAMMDDhhmmss																																
FimHorarioVerao	7	14	Data e hora do fim do horário de verão. Formato: AAAAMMDDhhmmss																																
Senha	8	6	Número da nova senha. Formato: 000000																																
Reiniciar	9	0	Reiniciar o REP.																																
HorarioAtual	10	14	Data e hora atual. Formato: AAAAMMDDhhmmss																																

Observações

Em Delphi, ao invés do enumerador, existem constantes que representam os valores acima. As constantes possuem o mesmo nome do enumerador, precedidas de “eParamSetConfig_”.

Por exemplo:

eParamSetConfig_EnderecolP

6.3. eParamGetConfig

Enumerador utilizado para indicar os valores possíveis para a leitura de configurações do REP.

Tipo de dado

Inteiro.

Valores possíveis

Código	Valor	Tamanho	Descrição																																
EnderecoIP	101	15	Endereço IP do REP. Formato: 000.000.000.000																																
PortaUDP	102	5	Porta UDP do REP (De 1 a 65536). Exemplo: 19001																																
MascaraRede	103	15	Máscara de rede do REP. Formato: 000.000.000.000																																
Roteador	104	15	Endereço do roteador do REP. Formato: 000.000.000.000																																
Tipoidentificacao	105	1	Tipo de identificação aceita pelo REP. Valores possíveis: <table border="1"> <thead> <tr> <th>Valor</th><th>Teclado</th><th>Leitor</th><th>Bio</th></tr> </thead> <tbody> <tr> <td>1</td><td>OK</td><td>NOK</td><td>NOK</td></tr> <tr> <td>2</td><td>NOK</td><td>OK</td><td>NOK</td></tr> <tr> <td>3</td><td>OK</td><td>OK</td><td>NOK</td></tr> <tr> <td>4</td><td>NOK</td><td>NOK</td><td>OK</td></tr> <tr> <td>5</td><td>OK</td><td>NOK</td><td>OK</td></tr> <tr> <td>6</td><td>NOK</td><td>OK</td><td>OK</td></tr> <tr> <td>7</td><td>OK</td><td>OK</td><td>OK</td></tr> </tbody> </table>	Valor	Teclado	Leitor	Bio	1	OK	NOK	NOK	2	NOK	OK	NOK	3	OK	OK	NOK	4	NOK	NOK	OK	5	OK	NOK	OK	6	NOK	OK	OK	7	OK	OK	OK
Valor	Teclado	Leitor	Bio																																
1	OK	NOK	NOK																																
2	NOK	OK	NOK																																
3	OK	OK	NOK																																
4	NOK	NOK	OK																																
5	OK	NOK	OK																																
6	NOK	OK	OK																																
7	OK	OK	OK																																
InicioHorarioVerao	106	14	Data e hora do início do horário de verão. Formato: AAAAMMDDhhmmss																																
FimHorarioVerao	107	14	Data e hora do fim do horário de verão. Formato: AAAAMMDDhhmmss																																
HorarioAtual	110	14	Data e hora atual. Formato: AAAAMMDDhhmmss																																
NumeroSerie	201	7	Número de série do REP. Formato: 0000000																																
Empregador	202	150	Razão social do empregador. Texto com até 150 caracteres.																																
Documento	203	14	Número do documento do empregador. Formato: 0000000000000000																																
Local	204	100	Local de prestação de serviços do empregador. Texto com até 100 caracteres																																
QtdeEmpregados	205	5	Quantidade de empregados cadastrados no REP. Formato: 00000																																

QtdeLancamentos	206	9	Quantidade de lançamentos na MRP do REP. Formato: 000000000
InicioOperacao	207	14	Data e hora do início de operação do REP. Formato: AAAAMMDDhhmmss
UltimoRegistro	208	14	Data e hora do último registro na MRP do REP. Formato: AAAAMMDDhhmmss
NsrAtual	209	9	NSR atual do REP. Formato: 000000000
RegistrosLivres	210	9	Quantidade de registros livres na MRP do REP. Formato: 000000000
TamanhoMRP	211	8	Tamanho, em Kb, da MRP utilizada no REP. Formato: 00000000
EspacoLivre	212	8	Espaço livre, em Kb, disponível na MRP do REP. Formato: 00000000
MacAddress	213	17	MAC Address do REP. Formato: 00-00-00-00-00-00
NumeroModelo	214	5	Número do modelo do REP. Formato: 00000
NumeroFabricante	215	5	Número do fabricante do REP. Formato: 00000
CEI	216	12	Número do CEI do empregador. Formato: 000000000000
TipoDocumento	217	1	Tipo de documento do empregador. Valores possíveis: 1 = CNPJ 2 = CPF
StatusPapel	218	1	Status da bobina de papel. Valores possíveis: 0 = Sem papel. 1 = Papel OK.
NumeroSerieCompleto	219	17	Número de série completo do equipamento, formado pelo número do fabricante (FFFFF) + número do modelo (MMMMM) + número de série (SSSSSS). Formato: FFFFFMMMMSSSSSSS

Observações

Em Delphi, ao invés do enumerador, existem constantes que representam os valores acima. As constantes possuem o mesmo nome do enumerador, precedidas de “eParamGetConfig_”.

Por exemplo:

eParamGetConfig_EnderecoIP

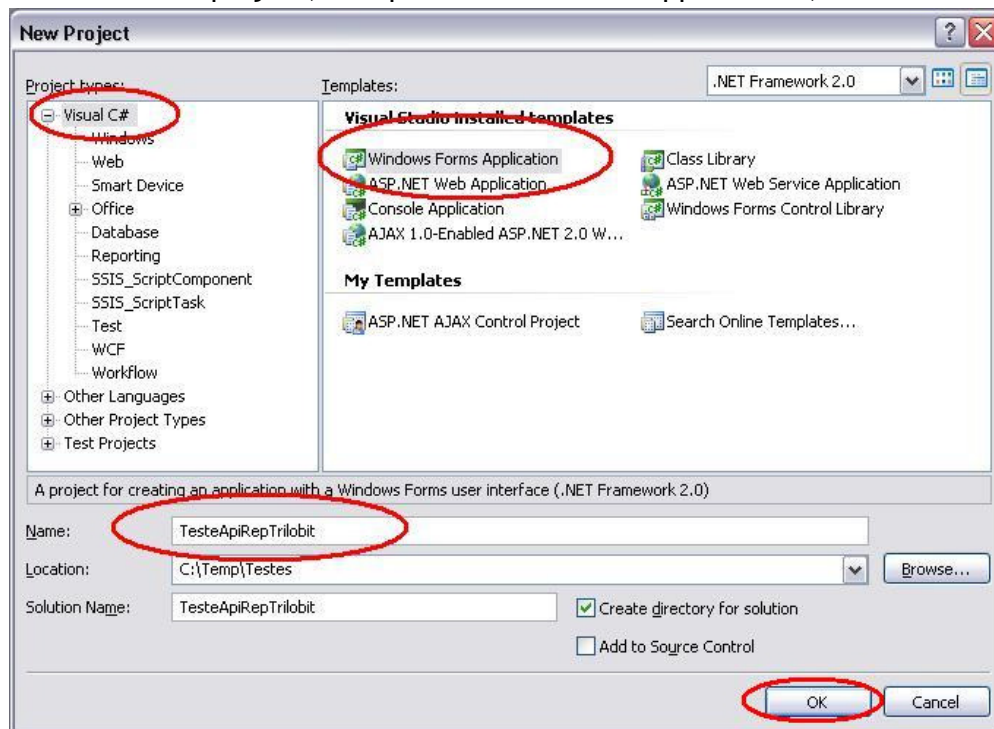
7. Formas de Utilização

Neste tópico veremos como utilizar a API RepTrilobit.

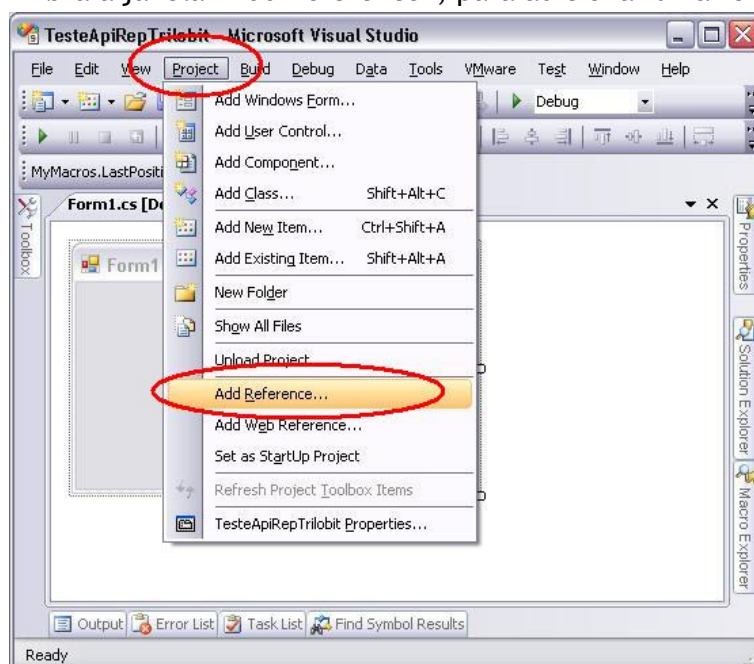
7.1. C# (Utilizando Visual Studio 2008)

Para utilizar a API RepTrilobit no C# com o Visual Studio 2008 execute os seguintes passos:

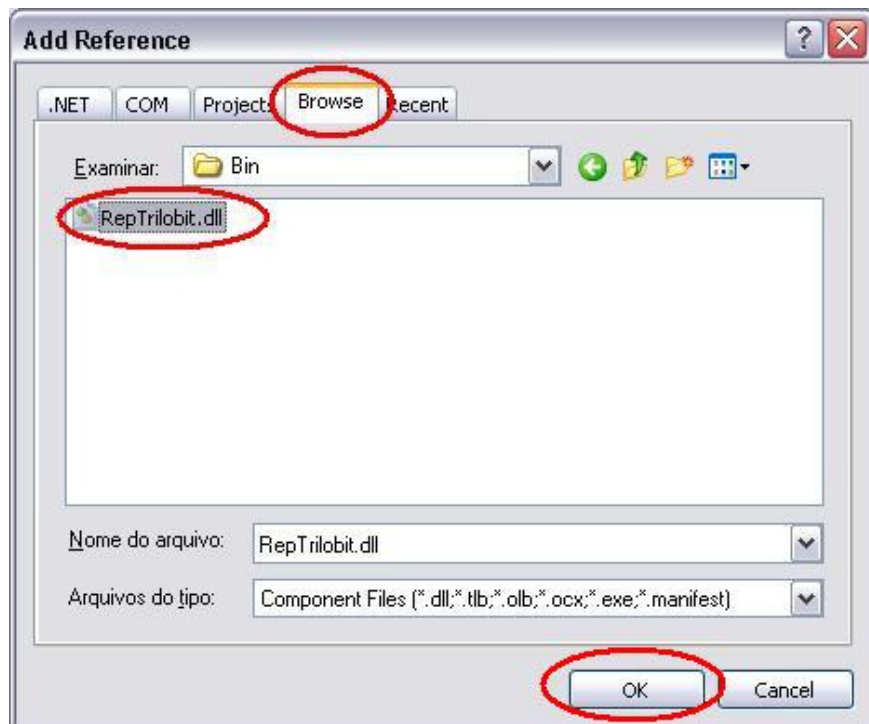
- Crie um novo projeto, do tipo “Windows Form Application”, utilizando “Visual C#”.



- Abra a janela “Add Reference”, para adicionar uma referência à API RepTrilobit ao projeto.



- Clique na aba "Browse", procure pelo arquivo RepTrilobit.dll, selecione-o e depois clique em "OK".



- Adicione um botão ao Form, dê um duplo clique para criar um novo tratador de eventos e inclua o seguinte código:

```
private void button1_Click(object sender, EventArgs e)
{
    RepTrilobit.REP meuREP = new RepTrilobit.REP();
    string ValorRetorno = "";
    if (meuREP.LerConfiguracao("127.0.0.1", 19001, 1,
        RepTrilobit.REP.eParamGetConfig.Empregador, ref ValorRetorno))
    {
        MessageBox.Show("Valor retornado=" + ValorRetorno);
    }
    else
    {
        MessageBox.Show("Ocorreu o seguinte erro: " + meuREP.ErrorException.Message);
    }
}
```

- Execute a aplicação. Caso exista um RepVirtual na mesma máquina em que o projeto está sendo testado, o nome do empregador deverá aparecer em uma mensagem na tela. Caso não exista um RepVirtual ou algum outro erro ocorra, aparecerá um mensagem com maiores detalhes.

7.2. Visual Basic 6

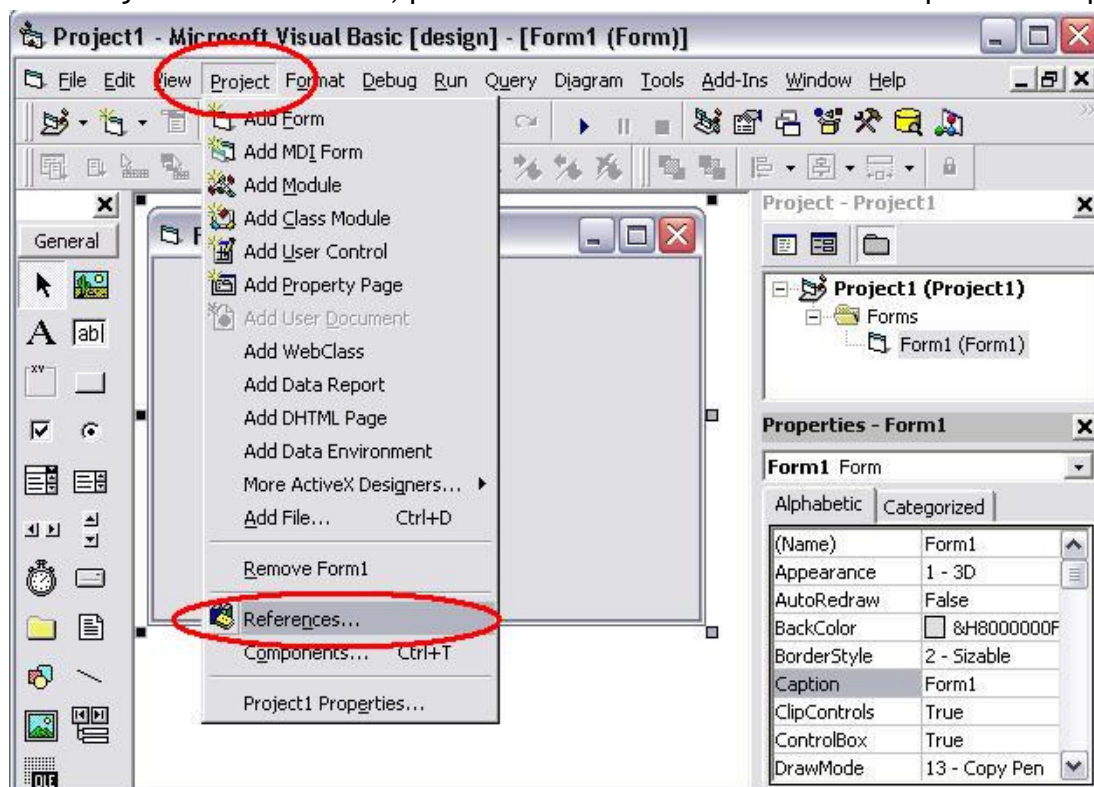
Para utilizar a API RepTrilobit no Visual Basic 6, é necessário que os arquivos RepTrilobit.dll e RepTrilobit.tlb tenham sido instalados e registrados de maneira adequada, através do Setup do SDK Rep Trilobit.

Para criar um projeto usando o Visual Basic 6 execute os seguintes passos:

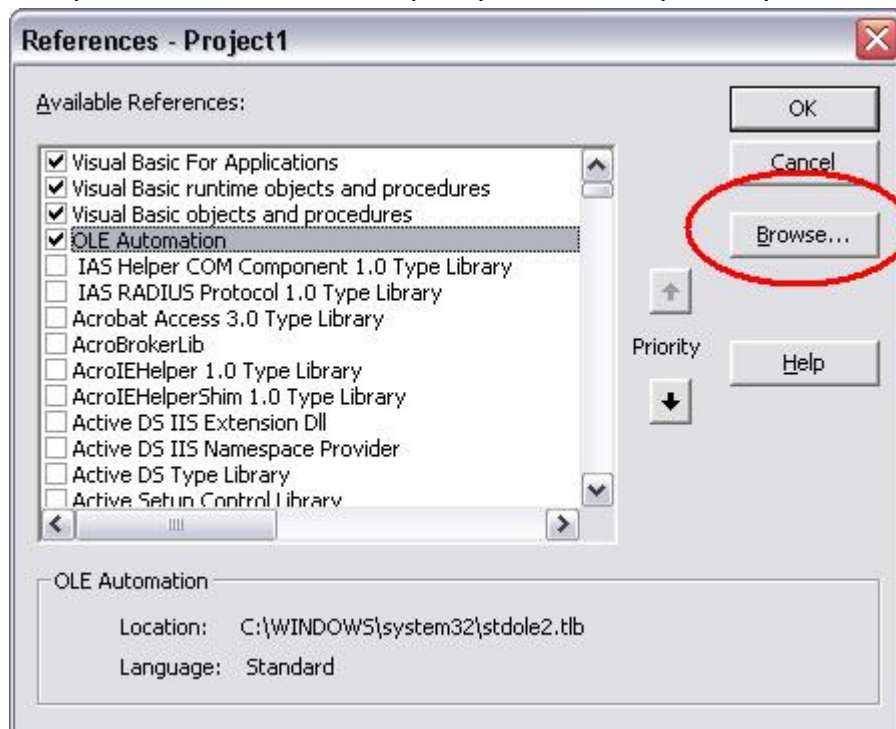
- Crie um novo projeto, do tipo “Standard EXE”



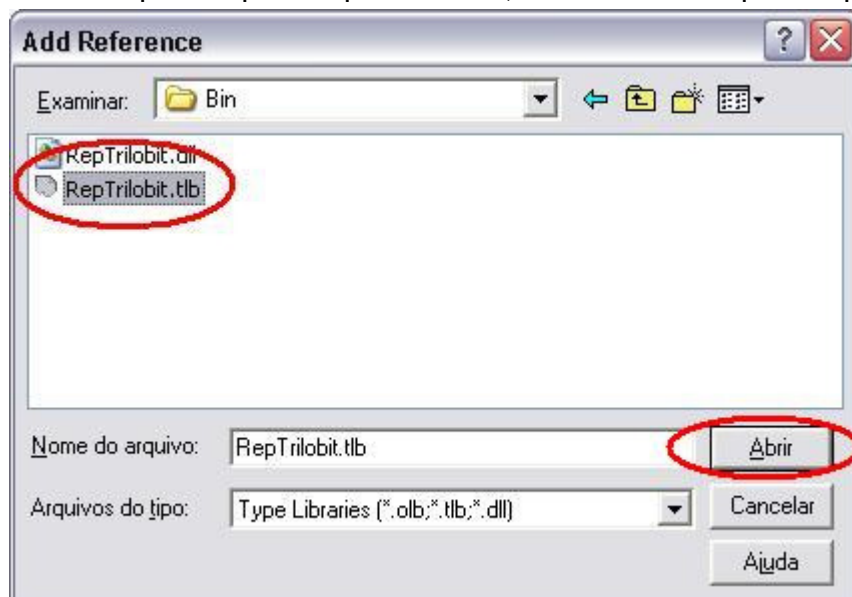
- Abra a janela “References”, para adicionar uma referência à API RepTrilobit ao projeto.



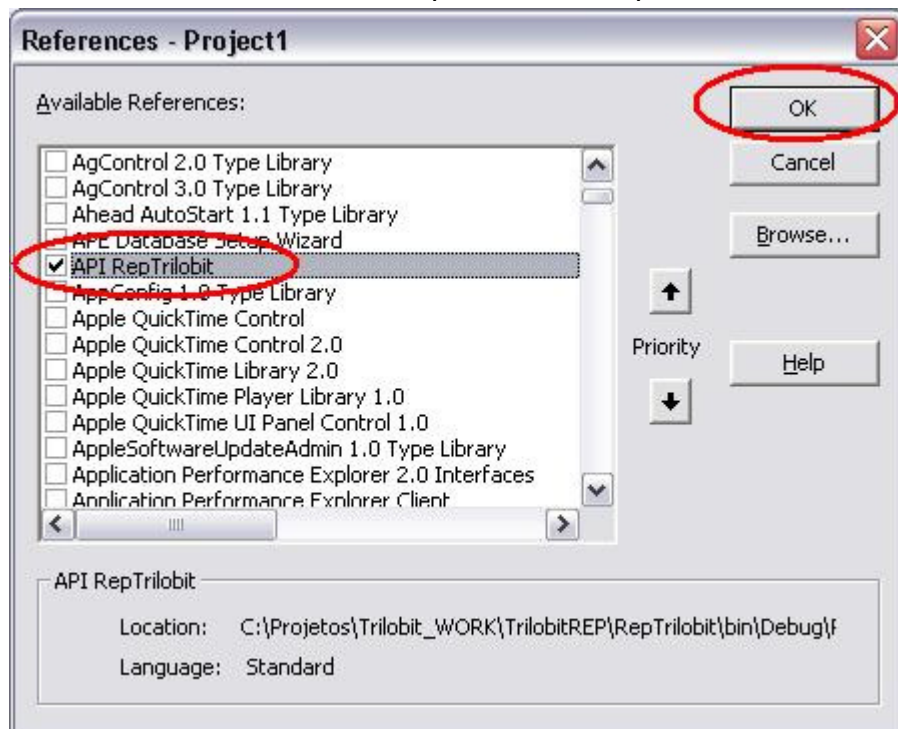
- Clique no botão “Browse...” para procurar o arquivo RepTrilobit.tlb.



- Procure pelo arquivo RepTrilobit.tlb, selecione-o e depois clique em “Abrir”.



- Selecione a referência “API RepTrilobit” e clique em “OK”.



- Adicione um botão ao Form, dê um duplo clique para criar um novo tratador de eventos e inclua o seguinte código:

```
Private Sub Command1_Click()  
  
    Dim meuREP As New RepTrilobit.REP  
    Dim ValorRetorno As String  
    If meuREP.LerConfiguracao("127.0.0.1", 19001, 1, _  
        eParamGetConfig_Empregador, ValorRetorno) Then  
        MsgBox "Valor retornado=" & ValorRetorno  
    Else  
        MsgBox "Ocorreu o seguinte erro: " & _  
            meuREP.Exception.Message  
    End If  
  
End Sub
```

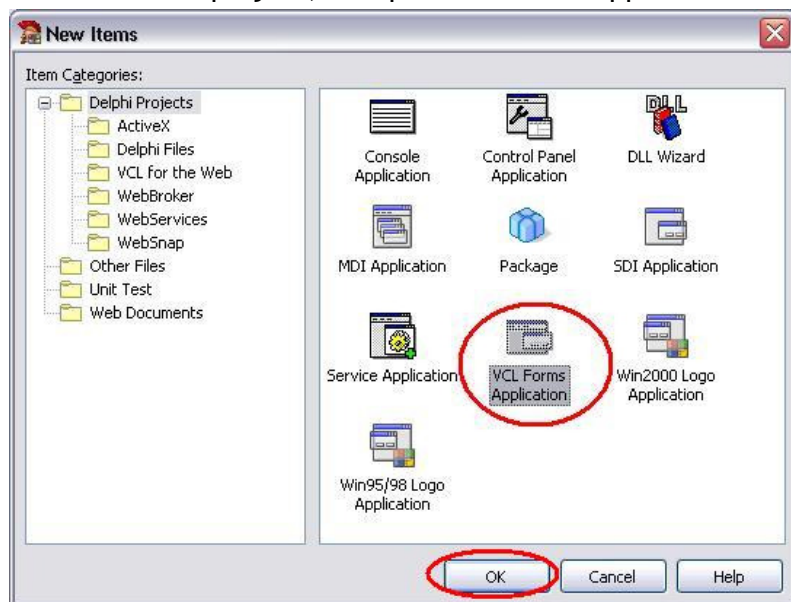
- Execute a aplicação. Caso exista um RepVirtual na mesma máquina em que o projeto está sendo testado, o nome do empregador deverá aparecer em uma mensagem na tela. Caso não exista um RepVirtual ou algum outro erro ocorra, aparecerá um mensagem com maiores detalhes.

7.3. Delphi (Utilizando Delphi 2007)

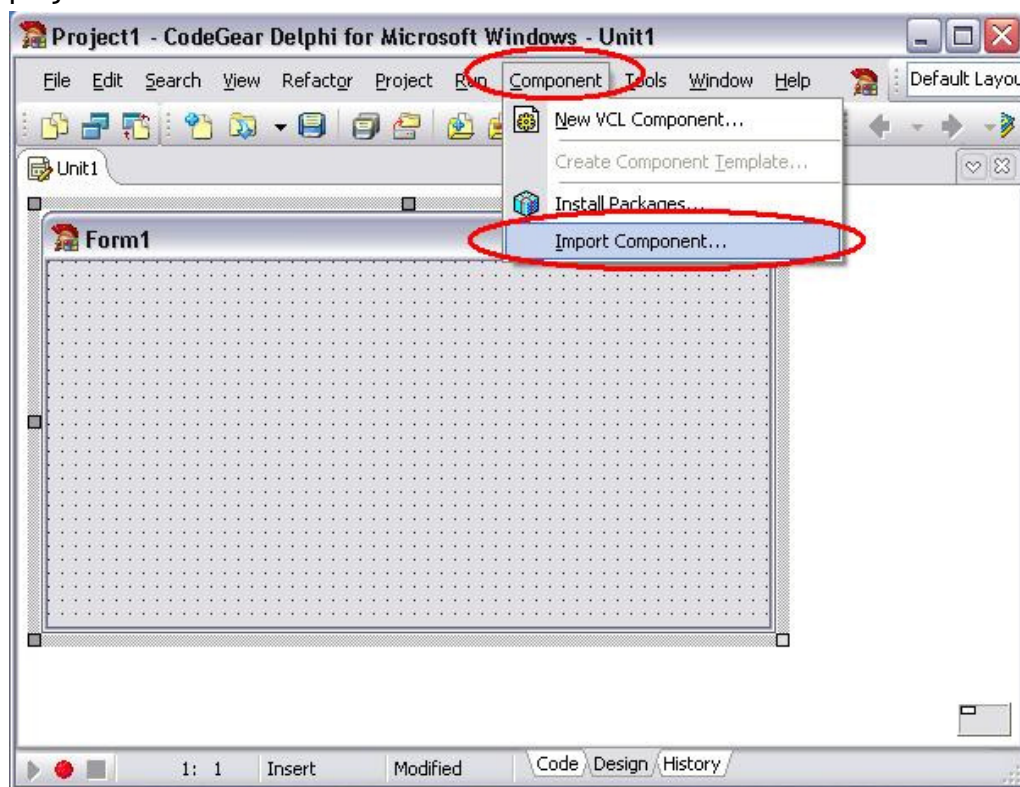
Para utilizar a API RepTrilobit no Delphi, é necessário que os arquivos RepTrilobit.dll e RepTrilobit.tlb tenham sido instalados e registrados de maneira adequada, através do Setup do SDK Rep Trilobit.

Para criar um projeto usando o Delphi 2007 execute os seguintes passos:

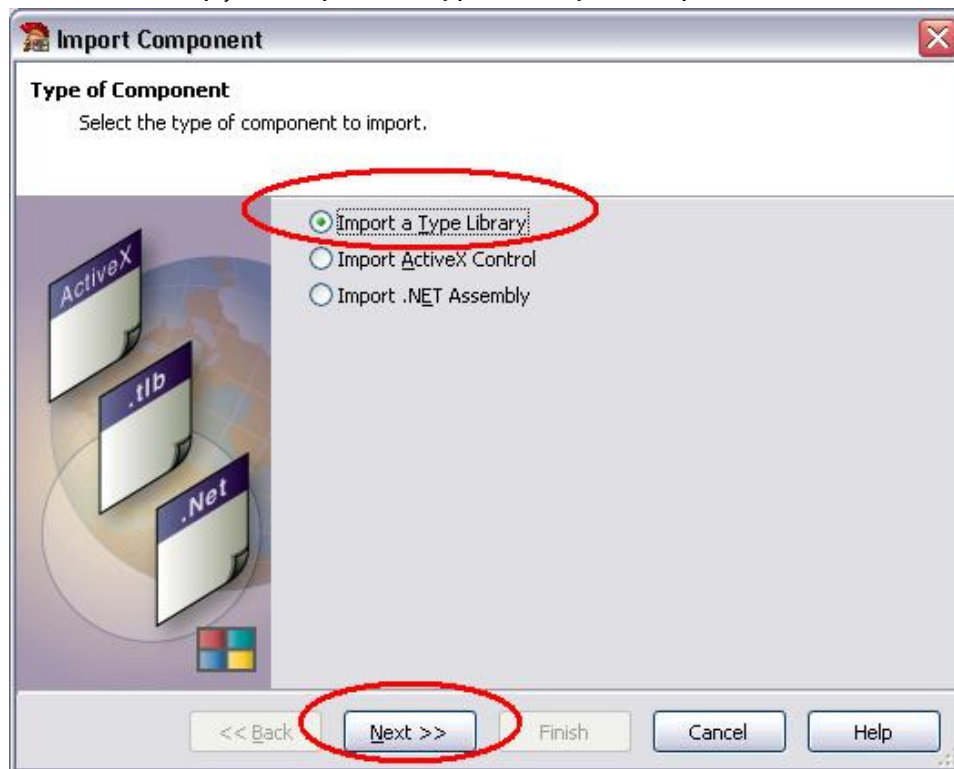
- Crie um novo projeto, do tipo “VCL Forms Application”



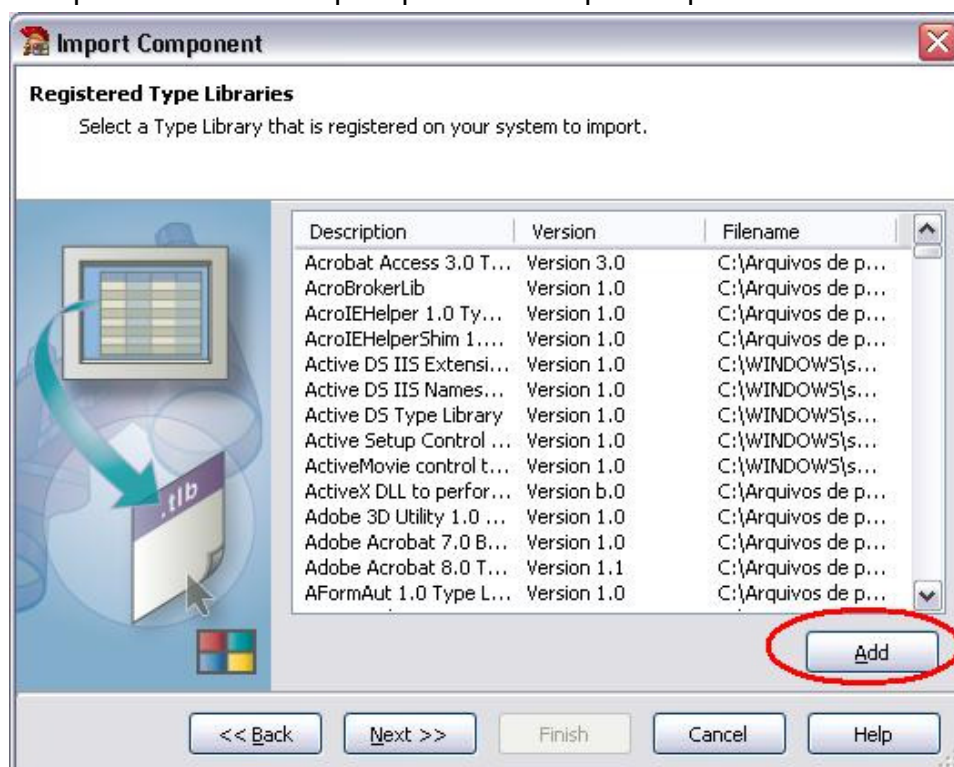
- Abra a janela “Import Component”, para adicionar uma referência à API RepTrilobit ao projeto.



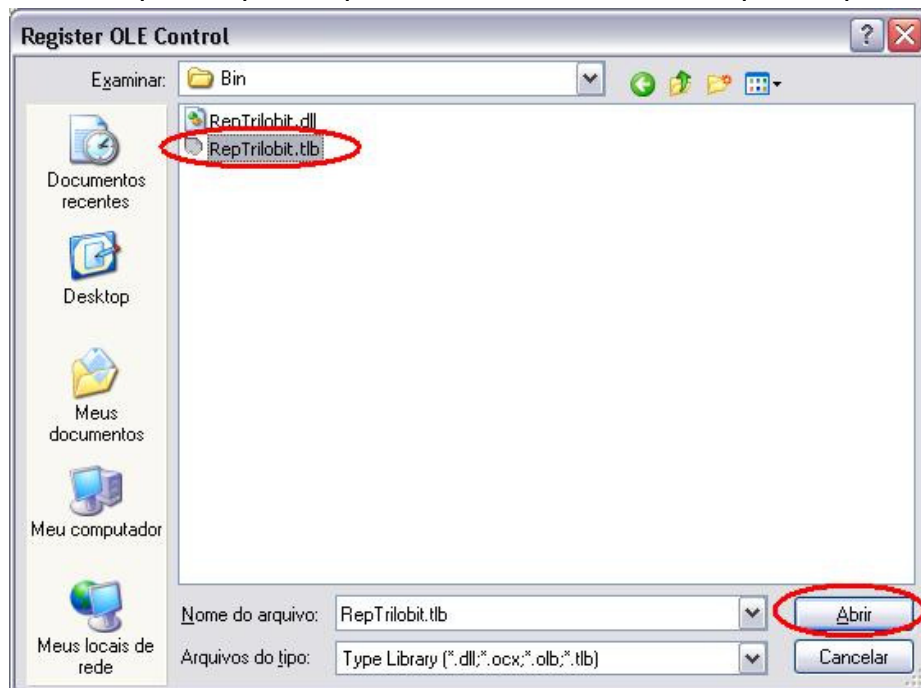
- Selecione a opção “Importa a Type Library” e clique em “Next”.



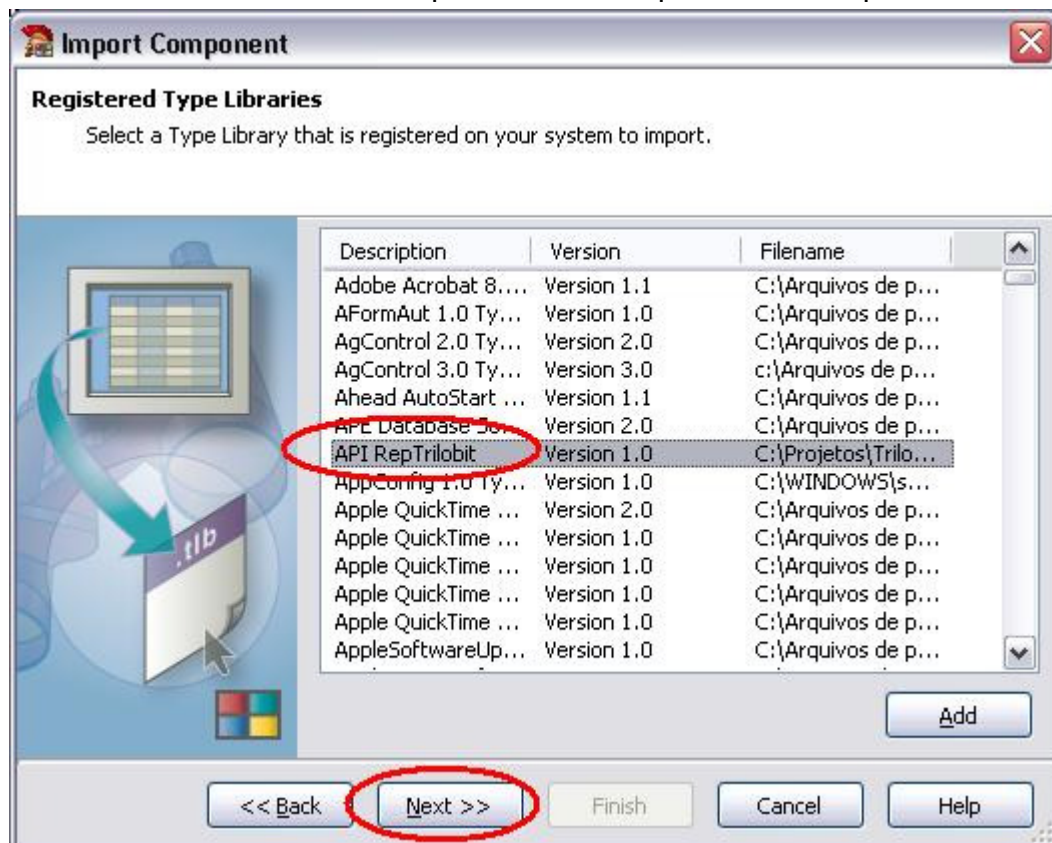
- Clique no botão “Add” para procurar o arquivo RepTrilobit.tlb.



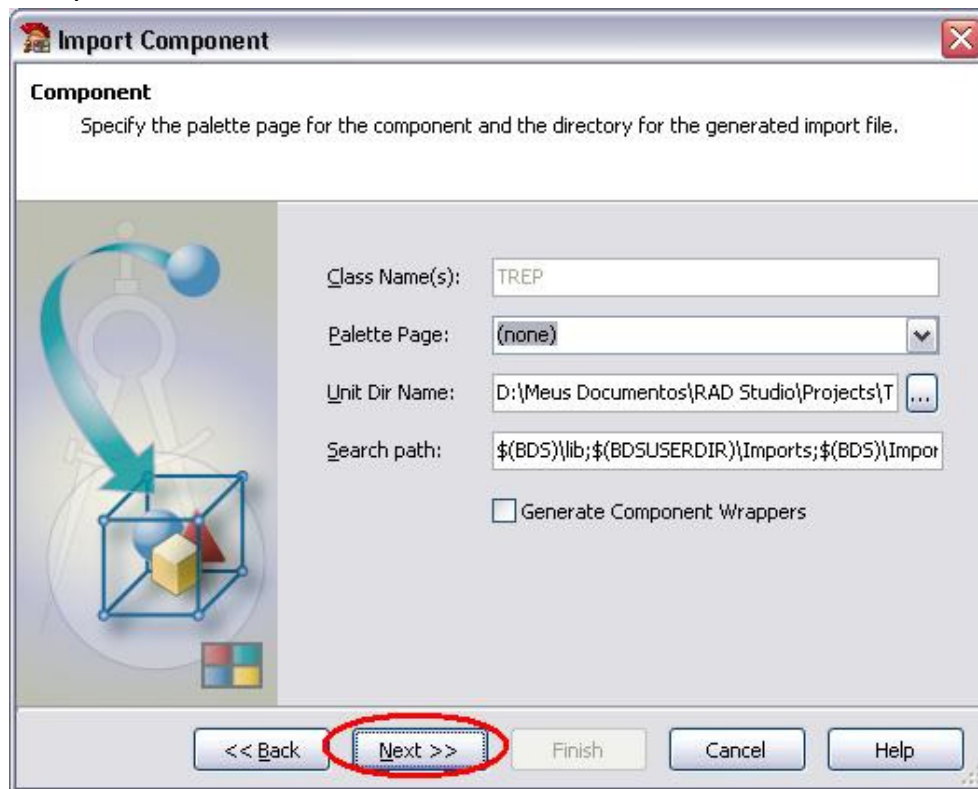
- Procure pelo arquivo RepTrilobit.tlb, selecione-o e depois clique em “Abrir”.



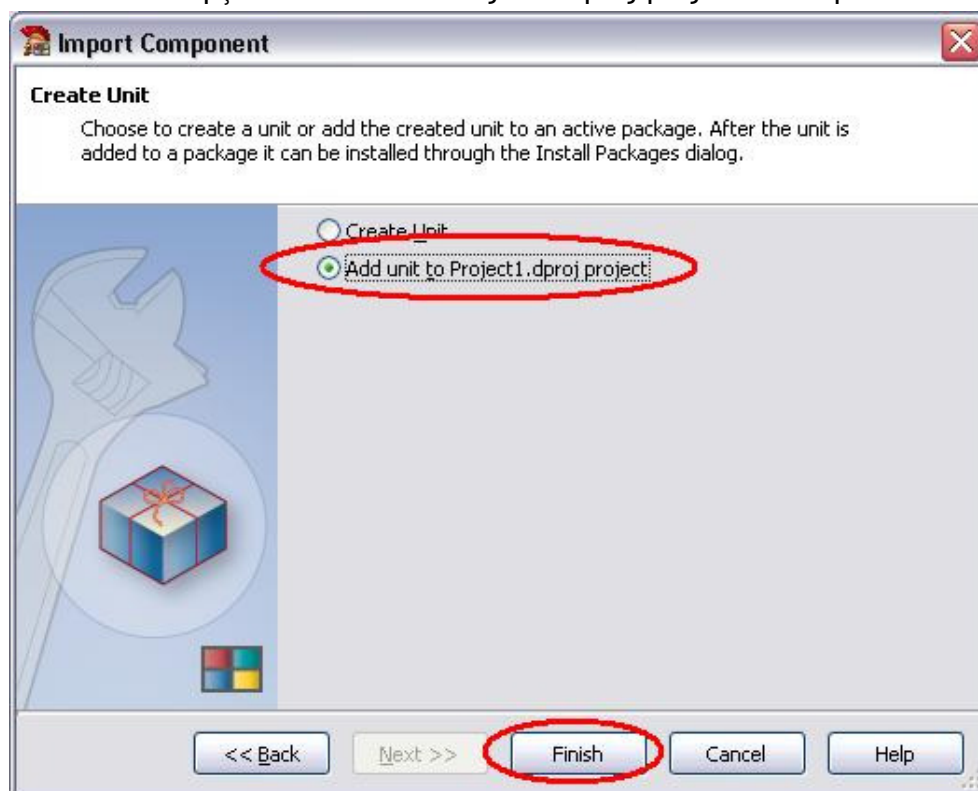
- Selecione a referência “API RepTrilobit” ou “RepTrilobit” e clique em “Next”.



- Clique em “Next”.



- Selecione a opção “Add unit to Project1.dproj project” e clique em “Finish”.



- Adicione a “RepTrilobit_TLB” ao “uses” do Form.

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, RepTrilobit_TLB;

- Adicione um botão ao Form, dê um duplo clique para criar um novo tratador de eventos e inclua o seguinte código:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    meuREP: REP;  
    ValorRetorno: WideString;  
begin  
    meuREP := CoREP.Create;  
    if meuREP.LerConfiguracao('127.0.0.1', 19001, 1,  
        eParamGetConfig_NumeroSerieCompleto, ValorRetorno) = true then  
        begin  
            ShowMessage('Valor retornado=' + ValorRetorno);  
        end  
    else  
        begin  
            ShowMessage('Ocorreu o seguinte erro: ' + meuREP.ErrorException.Message);  
        end;  
end;  
  
end.
```

- Execute a aplicação. Caso exista um RepVirtual na mesma máquina em que o projeto está sendo testado, o nome do empregador deverá aparecer em uma mensagem na tela. Caso não exista um RepVirtual ou algum outro erro ocorra, aparecerá um mensagem com maiores detalhes.

Manual do SDK para REP Trilobit
Módulo: API RepTrilobit

TRILOBIT

Rua Alvarenga, 1377 - CEP 05509-002
São Paulo / SP - Brasil
Fone: (11) 3032-4196
suporte@trilobit.com.br
<http://www.trilobit.com.br>