

Web Services com Protheus

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

O Protheus, a partir da versão AP7, possui ferramentas nativas e integradas com a LIB de Infra-Estrutura do ERP, para desenvolvimento de aplicações 'Cliente' e 'Server', utilizando a tecnologia dos Web Services. Para melhor compreensão do assunto, os tópicos relacionados a ambos foram didaticamente separados em Aplicações Server e Aplicações Cliente, respectivamente. Nos tópicos 'Comandos' e 'Funções', são abordadas respectivamente as diretivas e funções da Lib de Infra-estrutura do ERP disponibilizadas para o desenvolvimento de ambas as aplicações, Cliente e Server. No tópico 'Exemplos Advpl', são demonstrados os exemplos 'atômicos' de uso das funções e comandos.

COMANDOS - ENDWSCLIENT

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

ENDWSCLIENT [self]

Parâmetros

Argumento	Tipo	Descrição
self	(NULO)	Esta instrução não recebe nenhum parâmetro.

Descrição

Através desta instrução, encerra-se a declaração de uma classe 'Client' de Web Services, iniciada com o *statement* WSCLIENT.

Esta instrução de declaração é utilizada exclusivamente quando da geração de um fonte 'Cliente' de Web Services, através do assistente 'Gerar Cliente WebServices...' do IDE.

Observação : A utilização deste comando exige a declaração do *#include* 'APWEBSRV.CH' no fonte Advpl.

COMANDOS - ENDWSSERVICE

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

ENDWSSERVICE [self]

Parâmetros

Argumento	Tipo	Descrição
self	(NULO)	Esta instrução não recebe nenhum parâmetro.

Descrição

Através desta instrução, encerra-se a declaração de uma classe 'Server' de Web Services, iniciada com o *statement* WSSERVICE.

O não-fechamento da declaração da classe ocasiona "falha de compilação" no fonte.

Observação : A utilização deste comando exige a declaração do *#include* 'APWEBSRV.CH' no fonte Advpl.

COMANDOS - ENDWSSTRUCT

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

ENDWSSTRUCT [self]

Parâmetros

Argumento	Tipo	Descrição
self	(NULO)	Esta instrução não recebe parâmetros

Descrição

Através desta instrução, encerra-se a declaração de uma estrutura a ser utilizada em um Web Service, iniciada com o *statement* WSSTRUCT.

O não-fechamento da declaração da estrutura ocasiona falha de compilação no fonte.

Observação : A utilização deste comando exige a declaração do *#include 'APWEBSRV.CH'* no fonte Advpl.

COMANDOS - WSCLIENT

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

WSCLIENT cClientName

Parâmetros

Argumento	Tipo	Descrição
cClientName	Caracter	cClientName corresponde 'a classe do Web Service a ser gerada.

Descrição

Através desta instrução, inicia-se a declaração uma classe 'Cliente' de Web Services em Advpl. Esta instrução de declaração é utilizada exclusivamente quando da geração de um fonte 'Cliente' de Web Services, através do assistente 'Gerar Cliente WebServices...', disponível no Protheus IDE.

Para encerrar a declaração da classe, é utilizada a instrução ENDWSCLIENT.

Observação : A utilização deste comando exige a declaração do *#include 'APWEBSRV.CH'* no fonte Advpl.

COMANDOS - WSDATA

Revisão: 30/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

WSDATA cVarName AS [ARRAY OF] cVarType [OPTIONAL]

Parâmetros

Argumento	Tipo	Descrição
cVarName	Caracter	cVarName corresponde ao nome da propriedade a declarar.
AS	Caracter	Separador para indicar o tipo da propriedade.
ARRAY OF	Caracter	cVarType corresponde a um Tipo Soap / compatível de variável a ser utilizado no serviço. Veja os tipos suportados abaixo na Tabela A - Tipos de Dados
cVarType	Caracter	cVarType corresponde a um Tipo Soap / compatível de variável a ser utilizado no serviço. Veja os tipos suportados abaixo na Tabela A - Tipos de Dados
OPTIONAL	Caracter	Caso especificado , definimos que esta propriedade é opcional no contexto do Web Service .

Descrição

Utiliza-se esta instrução para declarar uma propriedade de uma classe para WebServices, 'Cliente' ou 'Server'.

Uma propriedade obrigatoriamente deve ter definida seu nome e tipo, e opcionalmente podemos definir que a mesma terá tratamento de array e/ou tratamento opcional.

Observação : A utilização deste comando exige a declaração do *#include 'APWEBSRV.CH'* no fonte Advpl.

COMANDOS - WSMETHOD

Revisão: 22/04/2004

Abrangência

Versão 7.10

Sintaxe

```
WSMETHOD cMethodName [ WSRECEIVE <param_in,...> ] [ WSEND  
<param_out> ] [ WSSERVICE <service_name> ]
```

Parâmetros

Argumento	Tipo	Descrição
cMethodName	Caracter	cMethodName corresponde 'ao nome do método do Web Service.
WSRECEIVE <param_in,...>	Caracter	Através desta instrução , declaramos quais são o(s) parametro(s) que este método recebe, separados por vírgulas. Caso um método não receba parâmetros , devemos declarar que o mesmo recebe o parâmetro reservado NULLPARAM.
WSEND <param_out>	Caracter	Através desta instrução , declaramos um e apenas um parâmetro de retorno de um Web Service .
WSSERVICE <service_name>	Caracter	cServiceName corresponde ao nome da classe do serviço ao qual o método atual pertence.

Descrição

Através desta instrução, incia-se a declaração de um método de um Web Service - 'Cliente' e/ou 'Server', em Advpl . Utilizamos esta instrução em dois momentos no desenvolvimento :

- Na declaração da classe 'Server' e/ou 'Cliente' do serviço.
- Na definição do fonte do método 'propriamente dito', do respectivo WebService.

Ao utilizarmos a instrução WSMETHOD dentro da declaração de uma classe WSSERVICE, informamos apenas o primeiro parâmetro (cMethodName) . Porém, ao declarar o fonte propriamente dito do método, todos os parâmetros desta instrução são obrigatórios.

Observação : A utilização deste comando exige a declaração do *#include 'APWEBSRV.CH'* no fonte Advpl.

COMANDOS - WSSERVICE

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

```
WSSERVICE cServiceName [ DESCRIPTION <cDescr> ] [ NAMESPACE <cClsNS> ]
```

Parâmetros

Argumento	Tipo	Descrição
cServiceName	Caracter	cServiceName corresponde ao nome do Serviço (Classe em Advpl) que será declarado / criado. A nomenclatura de um Web Service segue a regra de nomenclatura de funções Advpl .
DESCRIPTION <cDescr>	Caracter	cDescr corresponde à descrição do Serviço, mostrada na tela de índice de serviços, e fornecida também junto do WSDL gerado pelo servidor Protheus para o serviço especificado.
NAMESPACE <cClsNS>	Caracter	cClsNS corresponde ao NameSpace sob o qual este serviço deve ser publicado.

Descrição

Através desta instrução, iniciamos a declaração uma classe 'Server' de WebServices em Advpl.

Dentro da estrutura de uma Classe 'Server' de Web Services, devemos declarar os métodos disponibilizados da classe, e declaramos todas as propriedades , parâmetros e retornos utilizados por esta classe, devidamente especificadas, utilizando as instruções WSMETHOD e WSDATA, respectivamente.

Para encerrar a declaração da classe, utilizamos a instrução ENDWSSERVICE..

A declaração de uma classe 'Server' de Web Services deve ter a seguinte estrutura básica :

```
WSSERVICE <cSvcName> DESCRIPTION <cDescr> NAMESPACE <cClsNS>
```

```
WSDATA <xDataName> AS <xDataType>  
(... demais propriedades, parâmetros e retornos ...)
```


WSMETHOD <MethodName>
(... demais métodos da classe ...)

ENDWSSSERVICE

(... fonte(s) do(s) método(s)s desta classe ...)

Observação : A utilização deste comando exige a declaração do *#include* 'APWEBSRV.CH' no fonte Advpl.

COMANDOS - WSSTRUCT

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

WSSTRUCT cSructName

Parâmetros

Argumento	Tipo	Descrição
cSructName	Character	cStructName corresponde ao nome da estrutura a ser criada. Obedece 'as regras de nomenclatura de funções Advpl.

Descrição

Através desta instrução , iniciamos a declaração de uma estrutura , a ser utilizada por um Web Service 'Server', em Advpl . Dentro de uma estrutura, devemos apenas declarar as propriedades que a mesma contém, através da instrução WSDATA. Devemos finalizar a declaração da estrutura utilizando o comando ENDWSSTRUCT.

Observação : A utilização deste comando exige a declaração do *#include 'APWEBSRV.CH'* no fonte Advpl.

Exemplo de uso da função GETWSCERROR

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

No exemplo abaixo, é ilustrado o tratamento de erro sugerido para uma chamada de um método através de um programa 'Client', desenvolvido em Advpl.

```
#include 'Protheus.ch'
#include 'ApWebSrv.ch'

User Function TstService
Local oService , cSvcError , cSoapFCode ,cSoapFDescr

// Cria uma instância do serviço Cliente
oService := WSTeste():New()

// Realiza a chamada do método Hello() do serviço.
If oService:Hello()

    // Método executado com sucesso.
    MsgStop('Execução OK')

Else

    // Caso o método retorne .F. , devemos identificar e tratar a
    ocorrência

    cSvcError    := GetWSCError()           // Resumo do erro
    cSoapFCode   := GetWSCError(2)         // Soap Fault Code
    cSoapFDescr  := GetWSCError(3)         // Soap Fault Description

    If !empty(cSoapFCode)
        // Caso a ocorrência de erro esteja com o fault_code
        preenchido ,
        // a mesma teve relação com a chamada do serviço .
        MsgStop(cSoapFDescr,cSoapFCode)
    Else
        // Caso a ocorrência não tenha o soap_code preenchido
        // Ela está relacionada a uma outra falha ,
        // provavelmente local ou interna.
        MsgStop(cSvcError,'FALHA INTERNA DE EXECUCAO DO
SERVIÇO')
    Endif

Endif

oService := NIL

Return
```

Exemplo de uso da função GETWSCVER

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

No exemplo abaixo , obtemos a versão da Lib 'Cliente' de Web Services compilada no repositório atual.

```
User Function ShowVersions()  
Local cCliVers := GetWSCVer()  
MsgStop(cCliVers)  
Return
```

Exemplo de uso da função GETWSSVER

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

No exemplo abaixo , obtemos a versão da Lib 'Server' de Web Services compilada no repositório atual.

```
User Function ShowVersion()  
Local cSrvVers := GETWSSVER()  
MsgStop(cSrvVers)  
Return
```

Exemplo de uso da função SETSOAPFAULT

Revisão: 30/04/2004

Abrangência

Versão 7.10

Versão 8.11

No exemplo 01, partindo de um método de um WebServices 'Server', caso um parâmetro não atenda a faixa de dados necessária, o serviço retorna ao Client solicitante um *Soap-Fault*, indicando a ocorrência de erro.

No exemplo 02, retornamos um *Soap-Fault*, indicando que não estava disponível um recurso no servidor para o processamento requisitado. Neste, retornamos que o Fault Code é 'SOAPFAULT_RECEIVER', pois o pacote não foi processado não por ter algum conteúdo inválido, mas sim por alguma razão ligada ao ambiente do servidor.

Por default, o *Fault-Code* de um *Soap-Fault* é 'SOAPFAULT_SENDER', o que indica que o serviço não foi processado por alguma razão ligada ao pacote de dados enviados; e indica ao client que o pacote deve ser re-montado para que o serviço seja executado.

Exemplo 01

```
(...)  
If ::Indice > 1024  
    SetSoapFault('Argumento Inválido','O índice não pode ser maior que  
1024.')  
    Return .f.  
Endif  
(...)
```

Exemplo 02

```
(...)  
If !File('\extras\modelo.cfg')  
    SetSoapFault('Serviço Indisponível','',SOAPFAULT_RECEIVER)  
    Return .f.  
Endif  
(...)
```

Funções – GETWSCERROR

Revisão: 22/04/2004

Abrangência

Versão 7.10

Sintaxe

GETWSCERROR ([nInfo]) --> xErrorInfo

Parâmetros

Argumento	Tipo	Descrição
		nInfo especifica qual informação pertinente ao erro deve ser retornada, podendo ser :
nInfo	Numérico	1 - Retorna uma String contendo o Resumo do Erro COMPLETO (DEFAULT) 2 = Retorna uma String contendo o soap:fault_code , caso disponível . 3 = Retorna uma String contendo o soap:fault_String , caso disponível . 4 = Retorna um Objeto XML contendo os nodes completos com as informações do erro , apenas caso o erro seja um soap_Fault.

Retorno

Tipo	Descrição
(Qualquer)	Retorna a informação do erro solicitada através do parâmetro nInfo . Caso nInfo seja 1 , 2 ou 3 , o retorno é do tipo String . Caso seja tipo 4 , será retornado um Objeto XML.

Descrição

Utilizada no desenvolvimento de uma aplicação 'Client' de WebServices, através desta função é possível recuperar as informações pertinentes à uma ocorrência de erro de processamento de um método 'Client', após a execução do mesmo.

Caso a execução de um método 'Client' de Web Services retorne .F., deve ser utilizada a função GetWSError(), para identificar a origem da ocorrência. Durante uma operação de execução de um método 'Client' de WebServices, são possíveis ocorrências de erro das seguintes naturezas, em momentos específicos :

1 - Antes do pacote 'SOAP', com os parâmetros e dados pertinentes à requisição, ser enviado.

Durante a montagem do pacote SOAP, para envio dos parâmetros do método solicitado ao servidor, é realizada uma consistência do(s) parâmetro(s) a serem enviados, tais como a obrigatoriedade do parâmetro e o tipo *Advpl* com o qual o parâmetro foi alimentado. Se e somente se os parâmetros informados sejam válidos, o pacote SOAP montado é postado no servidor de WebServices.

2 - Ao postar o pacote 'SOAP' para o respectivo Webservice

Ao postar o pacote, caso o *host* do Web Service utilizado ou o servidor referente ao mesmo não foi localizado ou não esteja no ar.

3 - Após o envio do pacote e obtenção do devido retorno do Server.

Uma vez enviado ao Server, a interface client entra em modo 'stand-by', aguardando por um pacote de retorno SOAP do Server. Após a portagem, caso o pacote devolvido não esteja em conformidade com a declaração do serviço, ou o servidor devolveu um html ao invés de um xml 'SOAP'.

4 - Erro Interno de execução : Qualquer ocorrência de erro fatal, seja antes ou depois do envio da requisição, cuja origem não seja tratada ou prevista pelas rotinas 'Client' do Serviço, como por exemplo um retorno de um pacote XML com erro de sintaxe ou estruturalmente inválido .

Funções – GETWSCVER

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

GETWSCVER () --> cVersion

Retorno

Tipo	Descrição
Caracter	cVersion corresponde à versão do Build da Lib 'Cliente' de WebServices, compilada no repositório em uso atualmente.

Descrição

Utilizada no desenvolvimento de uma aplicação 'Cliente' de Web Services , através desta função é possível obter a string contendo a identificação da versão de Build da LIB de Infra-Estrutura do Web Services 'Cliente'.

Funções – GETWSSVER

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

GETWSSVER () --> cVersion

Retorno

Tipo	Descrição
Caracter	cVersion corresponde à versão do Build da Lib 'Server' de WebServices, compilada no repositório em uso atualmente.

Descrição

Utilizada no desenvolvimento de uma aplicação 'Server' de Web Services , através desta função é possível obter a string contendo a indentificação da versão de Build da LIB de Infra-Estrutura do Web Services 'Server'.

Funções – SETSOAPFAULT

Revisão: 30/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

SETSOAPFAULT (< cError > , < cString > , [nFCode] , [cFactor] , [cFDetail]) -->
.T.

Parâmetros

Argumento	Tipo	Descrição
cError	Caracter	Através de cError deve ser especificada uma descrição reduzida , referindo-se ao tipo do erro , por exemplo : Erro de argumento , Parametro Invalido , Falha de Arquivo ,...
cString	Caracter	Em cString deve-se especificar um detalhe maior da ocorrência , não exatamente um detalhe técnico , porém uma especificação objetiva da ocorrência. Por exemplo : Parametro XXXXX fora da faixa válida de dados , compreendida entre mmm e nnn , ...
nFCode	Numérico	Fault Code : Através deste parametro , é possível especificar a origem da ocorrência da Soap Fault . Segundo a documentação do SOAP, Versão 1.2 (publicada na W3C) , foram definidos 6 códigos de ocorrências standard de erro , detalhados na Tabela A. Caso não seja especificado , por default é assumido o código 5 (Sender)
cFactor	Caracter	Através de CFactor , é possível especificar explicitamente qual node / atributo do XML / Soap que não foi processado e/ou ocasionou a falha . Deve ser utilizado o formato anyURI (ref namespace http://www.w3.org/2001/XMLSchema) para especificar o node / atributo que ocasionou a falha.
cFDetail	Caracter	Através de cFDetail , é possível especificar para fins internos de processamento maiores detalhes sobre uma ocorrência de erro, especificamente relacionada ao processamento do corpo ("body") de um pacote SOAP.

Retorno

Tipo	Descrição
Lógico	Esta função sempre retorna .T. (true)

Descrição

Utilizada no desenvolvimento de uma aplicação 'Server' de WebServices, através desta função é possível setar uma ocorrência de erro tratada, referente à execução do serviço, ou impossibilidade de execução do método durante a execução do mesmo.

Dentre as razões pelas quais este tratamento é utilizado, podemos citar ocorrências relacionadas a validade dos dados, recebidos no pacote de parametros enviados pelo 'Cliente', como parâmetros invalidos ou fora da faixa de dados permitida pela rotina, ou ocorrências relacionadas ao 'Server', como a falta de um determinado recurso no server para o processamento, como uma falha de acesso a base de dados, ou qualquer outra razão implementada no serviço.

Tabela A - FAULT CODES

nFCode	Constante	Descrição
1	SOAPFAULT_VERSIONMISMATCH	NameSpace inválido encontrado no processamento do Soap:Body
2	SOAPFAULT_MUSTUNDERSTAND	Refere-se a falha de interpretação de um node / atributo contido no Soap:Header, especificado com o atributo mustUnderstand setado para 'true'
3	SOAPFAULT_DTDNOTSUPPORTED	A String Soap enviada como parâmetro continha um DTD (Document Type Definition).
4	SOAPFAULT_DATAENCODINGUNKNOWN	O HEader ou o Body do pacote SOAP está utilizando um encoding-type não suportado pelo server.
5	SOAPFAULT_SENDER	Refere-se a uma ocorrência de erro e/ou falha de processamento da ação, por algum tipo de inconsistência relacionada a falta de um ou mais dados necessários ao processamento. Indica uma ocorrência que requer que o pacote SOAP seja remontado para que seja realizada uma nova tentativa de acesso.

Refere-se a uma ocorrência de erro e/ou falha de processamento por razões que não estão especificamente relacionadas ao conteúdo do pacote SOAP e/ou parametros recebidos, porém relacionados 'a uma falha no Receptor do Serviço, como por exemplo o servidor estar bloqueado para manutenção. Este tipo de ocorrência não indica que existe falha no pacote enviado, mas cosuma-se utilizar para indicar uma ocorrência relacionada naquele instante de tempo ; possivelmente estando disponível posteriormente .

Observação : Para utilizarmos os mnemônicos, ao invés dos números, nos codigos de erro, precisamos declarar no fonte Advpl a utilização do Include 'ApWebSrv.ch'

Funções – SOAPDTGETD

Revisão: 22/04/2004

Abrangência

Versão 8.11

Sintaxe

SOAPDTGETD (< cDateTime >) --> dDate

Parâmetros

Argumento	Tipo	Descrição
cDateTime	Character String	no formato "Soap" DateTime, a ser considerada.

Retorno

Tipo	Descrição
Data	Retorna a data identificada na String cDateTime

Descrição

A partir de uma string Advpl, contendo uma data no formato '*soap' DateTime*', a função SoapDtGetD() retorna a data correspondente em Advpl, como um conteúdo do tipo '*D' Date*'.

Funções – SOAPDTGETT

Revisão: 22/04/2004

Abrangência

Versão 8.11

Sintaxe

SOAPDTGETT (< cDateTmie >) --> cTime

Parâmetros

Argumento	Tipo	Descrição
cDateTmie	Character	String, no formato "Soap" DateTime, a ser considerada.

Retorno

Tipo	Descrição
Character	Retorna o horário identificado, no formato HH:MM:SS

Descrição

A partir de uma string Advpl, contendo uma data no formato '*soap' DateTime*', a função SoapDtGetD() retorna o horário correspondente em Advpl, como um conteúdo do tipo '*C' Character*', no formato HH:MM:SS

Funções – SOAPDTMOUNT

Revisão: 22/04/2004

Abrangência

Versão 8.11

Sintaxe

SOAPDTMOUNT (< dData > , < cTmie >) --> cDateTmie

Parâmetros

Argumento	Tipo	Descrição
dData	Data	Data a ser considerada para a montagem do 'DateTime'
cTmie	Caracter	Horário, no formato hh:mm:ss, a ser considerado, para a montagem do 'DateTime'

Retorno

Tipo	Descrição
Caracter	String 'SOAP', correspondendo à Data e Horários especificados, no formato DATETIME.

Descrição

A partir de uma Data em Advpl , e um horário, especificado como string, a função SoapDtMount() retorna a data e horário especificados como uma string, no formato 'Soap' *DateTime*.

Funções – WSCLASSNEW

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

WSCLASSNEW (< cSrvStruct >) --> oNewStruct

Parâmetros

Argumento	Tipo	Descrição
cSrvStruct	Caracter	Especifique o nome da estrutura "Server" de Webservices para a criação do Objeto.

Retorno

Tipo	Descrição
Objeto	A função retorna uma referência à uma nova instância da estrutura passada como parâmetro. Caso a estrutura não exista, a função retornará NIL.

Descrição

Através da função WSClassNew(), é possível criar uma nova instância de uma estrutura (WSSTRUCT) de WebServices, criada para ser utilizada como uma estrutura 'Server'. A utilização desta instância, para criar instâncias de uma estrutura usada numa aplicação 'Server' de WebServices em AdvPl, evita a necessidade de criação de um método 'NEW' para cada estrutura.

Observação : Embora seja possível, não se deve utilizar esta instrução para inicializar uma estrutura criada em um fonte 'Client' em Advpl; pois as estruturas client possuem as definições do método NEW() de cada uma, com as devidas inicializações de parâmetros inerentes ao serviço.

Funções – WSDLDBGLEVEL

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

WSDLDBGLEVEL (< nLevel >) --> NIL

Parâmetros

Argumento	Tipo	Descrição
nLevel	Numérico	Através de nLevel , definimos qual o nível de informações a ser mostrado : 0 (default) = sem informações adicionais , 1 = Apenas pacote de retorno e 2 = Informações e pacote de Envio e Retorno . Obs: Devemos chamar esta função apos inicializado o Objeto 'Cliente' do Web Service.

Retorno

Tipo	Descrição
(NULO)	Esta função sempre retorna NIL

Descrição

Utilizada para depuração de uma aplicação 'Cliente' de Web Services em Advpl .
Através desta função, é possível setar, em tempo de execução, um 'echo' de informações adicionais pertinentes à execução de um método 'Client' de Web Services , a ser mostrado no console do servidor Protheus (caso habilitado) , permitindo ainda parametrizar um nível de detalhamento das informações a serem mostradas.

Observações

- O valor informado na chamada desta função, será mantido e considerado por todos os métodos de serviços 'Client' em Advpl, executados a partir de então nesta *Thread*, até que a aplicação seja finalizada, ou esta função seja chamada novamente.
- Esta função deve ser utilizada única e exclusivamente para fins de depuração, pois a mesma onera a *performance* da aplicação 'Client'.

Aplicações 'Server' em Advpl

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Neste tópico, e posteriores documentos, são detalhadas as atribuições e funcionalidades de uma aplicação 'Server' de Web Services, utilizando o Protheus, desde a criação da aplicação até as configurações necessárias para a publicação do Web Service.

A criação de um 'Server' de Web Services em Advpl consiste na montagem de uma classe Advpl especial, chamada **WSSERVICE**, onde cada método da classe é uma ação do Web Service. Caberá aos fontes desta classe o processamento de uma requisição e a geração do respectivo retorno, cabendo então à Lib de Web Services da Infra-Estrutura do ERP a camada de troca de dados, recepção, pré-validação e tratamento do pacote SOAP, as conversões de dados cabíveis do XML para as propriedades de parâmetro da classe Advpl e a montagem do pacote SOAP a partir das propriedades de retorno setadas pelo método executado e retorná-lo ao 'Client' solicitante do processamento, além de prover ao 'Cliente' o documento WSDL referente à(s) classe(s) 'Server' compilada(s) no repositório de objetos em uso e configurado para atender às requisições de processamento.

Este método de programação em camadas permite encapsular os tratamentos internos, em se tratando de protocolo HTTP, SOAP e WSDL, tornando relativamente fácil a missão de criar um Web Service 'Server' utilizando-se do Protheus. Basta escrever uma classe que receba nenhum, um ou mais que um parâmetro e devolva obrigatoriamente um retorno; configurar o Protheus Server para habilitar a interface HTTP e os Web Services, que a Lib faz todo o resto.

01. WebServices 'Server' - Configuração

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

O Servidor Protheus como 'SERVER' de WebServices

Um WebService em Advpl utiliza-se de Working Threads (***) para atender as solicitações de processamento através do protocolo HTTP.

Existem duas maneiras de habilitar o WebService : Através da criação da seção [WEBSERVICES] no arquivo de configuração do servidor, ou através da configuração manual de um ambiente de Working Threads Extended (WEBEX), também no inicializador. A diferença entre ambas é que a segunda opção permite especificar maiores detalhes do ambiente de execução do serviço, permite a configuração de serviços e Web Sites simultaneamente, e também atendimento diferenciado de processamento para mais de um host e diretórios virtuais. Quando utilizamos o Protheus 8, devemos utilizar o novo assistente de configuração do servidor Protheus - MP8WIZARD, para instalar e configurar o módulo de WebServices.

Segue abaixo um exemplo documentado de como configurar o servidor Protheus para WebServices, utilizando a chave [WEBSERVICES].

Observação : Esta configuração exige que a seção HTTP não esteja configurada no servidor Protheus. Esta configuração irá internamente habilitar o serviço de HTTP e configurar o processo de resposta para WebServices.

[WEBSERVICES]

Enable=1 ; (Obrigatório) Indica se o service está habilitado (1) ou não (0).

Environment=ENVTESTE ; (Obrigatório) Indica qual environment do Server que irá atender as requisições

Conout=0 ; (Opcional) Permite a exibição de informações dos status internos do serviço (default = 0 : desabilitado) . Utilizado APENAS para depuração, em casos específicos, pois prejudica significativamente a performance do(s) serviço(s).

Trace=0 ; (Opcional) Habilita a gravação de um arquivo de log (wsstrace.log), contendo as informações sobre todas as chamadas e status do Web Service (default = 0)

PrepareIn=01,01 ; (Obrigatório) Permite especificar qual a empresa e filial do ERP serão utilizados para a montagem do ambiente de processamento das requisições.

NameSpace = <http://localhost> ; (Opcional) Permite especificar o nome do namespace 'default', utilizado pelo(s) serviço(s) compilado(s) sem a definição de 'NameSpace'. (default = host atualmente utilizado)

URLLocation = <http://localhost> ; (Opcional) Permite especificar a url

responsável pelo atendimento às solicitações de processamento do(s) serviço(s) (default = host atualmente utilizado)

Para configurar o WebService manualmente, deve ser inicialmente habilitado o serviço de HTTP do servidor Protheus, configurar um processo WEBEX, apontando para funções internas de processamento dos Web Services, e configurar um host através do qual as requisições de processamento serão atendidas. Veja no exemplo abaixo :

```
[HTTP] ;; Configuração do protocolo HTTP
Enable=1
Port=80
Path=c:\Ap7\Http
```

```
[localhost] ;; A título de exemplo, configuramos o host da estação local.
Defaultpage=wsindex.apw
ResponseJob=WSTESTE
```

```
[WSTESTE] ; Configuracao do job para atender aos WebServices
TYPE=WEBEX ;; ( Obrigatório ) Tipo do Job para Web Services deve ser
WEBEX
```

```
ONSTART=__WSSTART ;; ( Obrigatório ) configuração fixa para Web
Services
```

```
ONCONNECT=__WSCONNECT ;; ( Obrigatório ) configuração fixa para
Web Services
```

```
Environment=ENVTESTE ;; Especifique qual ambiente (environment) do
servidor Protheus que irá atender aos WebServices.
```

```
INSTANCES=2,5 ;; ( Obrigatório ) Indica qual a quantidade minima (default )
e máxima de processos ( Threads ) que serão colocados na memória para atender às
solicitações de processamento do(s) serviço(s) publicado(s).
```

```
Conout=0 ;; ( Opcional ) Permite a exibição de informações dos status internos
do serviço ( default = 0 : desabilitado ) . Utilizado APENAS para depuração, em casos
específicos, pois prejudica significativamente a performance do(s) serviço(s).
```

```
Trace=1 ;; (Opcional) Habilita a gravação de um arquivo de log ( wsstrace.log ),
contendo as informações sobre todas as chamadas e status do Web Service ( default = 0
)
```

```
PrepareIn=01,01 ; (Obrigatório) Permite especificar qual a empresa e filial do
ERP serão utilizados para a montagem do ambiente de processamento das requisições.
```

```
NameSpace = http://localhost/ ;; ( Opcional ) Permite especificar o nome do
namespace 'default', utilizado pelo(s) serviço(s) compilado(s) sem a definição de
'NameSpace'. ( default = host atualmente utilizado )
```

```
URLLocation = http://localhost/ ;; ( Opcional ) Permite especificar a url
responsável pelo atendimento às solicitações de processamento do(s) serviço(s) ( default
= host atualmente utilizado )
```

WSINDEX - Índice de Serviços

Uma vez habilitada a configuração para Web Services, obtemos o acesso a uma interface HTTP de consulta ao índice de serviços publicados. Para tal, basta re-iniciar o servidor Protheus após a configuração ser realizada, abrir um Web Browser (por exemplo, o Internet Explorer), e acessar o link <http://<servidor>/wsindex.apw> . No caso

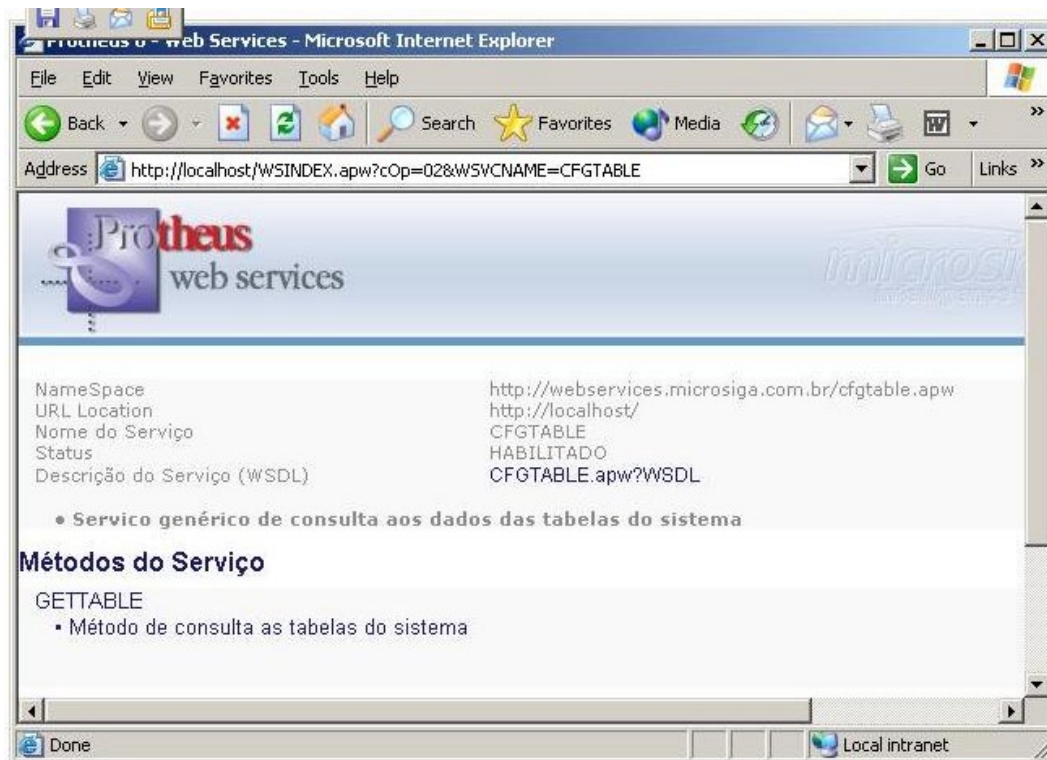
do exemplo de configuração acima, basta digitarmos <http://localhost/wsindex.apw> , e nos será apresentada a interface de consulta ao índice dos serviços.

Por exemplo, caso o host configurado para os wehservices for o host local (localhost) , devemos acessar o link <http://localhost/wsindex.apw> . Utilizando o Protheus8, será mostrada uma tela semelhante à vista abaixo:



Nesta interface são mostrados todos os serviços compilados e disponibilizados no repositório de objetos em uso no ambiente configurado. Através dela, é possível obter maiores detalhes sobre cada um dos serviços compilados.

Cada serviço ativo é um link para uma página, onde são mostrados todos os métodos do serviço, e onde é apresentado também um link através do qual o servidor Protheus fornecerá a descrição do serviço (WSDL). Logo abaixo é mostrado o exemplo da tela de detalhes do serviço CFGTABLE.



O Link para a obtenção do WSDL encontra-se acima em 'CFGTABLE.apw?WSDL'. Basta clicar neste link , que uma nova janela do Browser será aberta, mostrando o documento WSDL deste serviço.

Cada método do serviço disponibilizado também é um link, para uma página onde são mostrados os exemplos de pacotes SOAP que este método especificamente espera para recepção de parâmetros, e o modelo do pacote de retorno do serviço.

Caso o fonte-Client Advpl deste serviço seja gerado e esteja compilado no repositório atual, a interface de consulta habilita a funcionalidade de teste do Webservice, através da interface http, mostrando no final da tela o botão "testar". Ao clicar neste, é montada uma tela em HTML para que os parâmetros do serviço sejam preenchidos. Após os parâmetros preenchidos e submetidos, o pacote de retorno do serviço e seu respectivo status é retornado no Browse.

02. Criando um WebService 'Server' com o Protheus

Revisão: 23/04/2004

Abrangência

Versão 7.10

Versão 8.11

Para criarmos um WebService 'Server' utilizando o Protheus, primeiro devemos habilitar o servidor Protheus como servidor de WebServices. Para tal, veja o documento 'configurando o servidor Protheus para WebServices. '

Uma vez configurado e habilitado os WebServices no servidor Protheus, deve ser inicialmente determinados os métodos aos quais o serviço se destina; para então determinar os parâmetros e retorno de cada método. Uma vez determinadas estas informações, deve ser codificada uma classe especial em Advpl , chamada WSSERVICE, que constituirá o serviço propriamente dito.

Porém, antes de partir para a codificação, é fortemente recomendado que sejam lidos os documentos deste tópico, onde são abordados em detalhes a infra-estrutura envolvida com os WebServices, seu funcionamento e as particularidades de comportamento da classe de WebServices.

03. Regras para codificação de um WebService

Revisão: 30/04/2004

Abrangência

Versão 7.10

Versão 8.11

Visão Geral

Para a codificação de um webservice, foram criadas em Advpl instruções especiais de declaração de classe, específicas para WebServices, que suportam nomes 'longos' no nome da classe, métodos e propriedades. A utilização destes comandos exige a declaração do `#include 'apwebsrv.ch'` no topo do código-fonte; e exige também a atenção em alguns pontos e particularidades, a iniciar pela nomenclatura do serviço, estruturas, métodos e propriedades.

Características operacionais do ambiente

Devemos estar atentos ao desenvolver os métodos de WebServices, devido às características operacionais do ambiente de 'Working Threads' utilizado pelo Web Services. Ao executar um método do WebServices, o ambiente será mantido no ar, aguardando uma nova requisição de processamento, de qualquer serviço ou método, e de qualquer cliente. De modo que, ao desenvolver um serviço, não devemos deixar abertos as "Queries" utilizadas no método, filtros setados em tabelas principais, ou configurações específicas não-padrão do ambiente, realizadas para o processamento de um método específico; pois isto pode causar impacto no funcionamento de todos os WebServices compilados e ativos neste servidor, com efeitos imprevisíveis.

Nomenclatura dos Serviços

O nome de uma classe para WebServices, deve ser iniciado por um caractere alfabético, e deve conter apenas os caracteres alfabéticos compreendidos entre A e Z, os caracteres numéricos compreendidos entre 0 e 9, podendo também ser utilizado o caracter “_” (underline) . Um serviço não pode ter um nome de uma palavra reservada Advpl, e não pode ter o nome igual a um tipo básico de informação.

Nomenclatura de Estruturas

O nome dado à uma estrutura obedece as mesmas regras de nomenclatura de Serviços; não podendo haver uma estrutura com o mesmo nome de um serviço declarado. Devemos estar atentos também ao fato de uma estrutura não estar diretamente ligada ao serviço em questão, de modo que não podemos compilar duas estruturas de mesmo nome no mesmo repositório.

Uma estrutura constitui um agrupamento de dados, criado como uma classe especial (WSSTRUCT) em Advpl. Devemos criar de uma estrutura para um serviço, quando é necessário agrupar um conjunto de dados básicos e/ou outras estruturas em um único tipo de informação, que será utilizada como parâmetro e/ou retorno em um ou mais métodos do serviço.

Nomenclatura das propriedades - parâmetros e retorno

Cada parâmetro e retorno de todos os métodos de um serviço devem ser declarados como uma propriedade da classe do Serviço. Para dar nome a estes, são válidas as mesmas regras de nomenclatura de Serviços, não podendo haver um dado com o mesmo nome de um serviço ou estrutura já declarados anteriormente.

04. Tipos Básicos de Dados - 'Server'

Revisão: 23/04/2004

Abrangência

Versão 7.10

Versão 8.11

Quando escrevemos um Webservice 'Server', devemos especificar o tipo da informação de cada parâmetro e retorno, em conformidade com a especificação 'SOAP', utilizada nos pacotes XML de troca de dados.

São considerados e suportados pelo Protheus, quando da declaração dos parâmetros e retorno, os seguintes tipos básicos :

String	Dado Advpl do tipo String.
Date	Dado Advpl do tipo Data.
Integer	Dado Advpl do Tipo numérico (apenas numeros inteiros.)
Float	Dado Advpl do Tipo numérico (pode conter numeros inteiros e não-inteiros.)
Boolean	Dado Advpl do Tipo Booleano (lógico) .
Base64Binary	Dado Advpl do Tipo String Binária , aceitando todos os Caracteres da Tabela ASCII , de CHR(0) a CHR(255)

Observações

- Ao declararmos uma propriedade como sendo do tipo "String", não podemos especificar a palavra "String" em letras maiúsculas. A palavra STRING, escrita desta maneira, é interpretada pelo pré-compilador do Protheus como sendo uma constante, ocasionando erro de sintaxe da compilação do Webservice.

05. Estruturas - Tipos complexos

Revisão: 23/04/2004

Abrangência

Versão 7.10

Versão 8.11

Definição de Estrutura

Uma estrutura (também conhecida por *Complex Type*), constitui uma classe especial do Advpl, chamada WSSTRUCT, criada especificamente para WebServices. Devemos criar uma estrutura quando temos a necessidade de agrupar mais de uma informação, incluindo tipos básicos e/ou outras estruturas.

Ao criarmos um serviço que deverá receber como parâmetro um grupo de informações definido, por exemplo, os dados cadastrais de um cliente, devemos criar uma estrutura para agrupar estes dados. Vale ressaltar que a declaração de uma estrutura não amarra a mesma ao serviço em questão, de modo que a mesma estrutura pode ser utilizada para mais de um serviço compilado no repositório. Caso a estrutura criada seja específica para o serviço em questão, é recomendado que seja dado um nome à mesma que etnha a ver com o serviço ao qual ela pertença, pois não é possível compilar mais de uma estrutura de mesmo nome no repositório.

06. Métodos 'Server' em Advpl - Características

Revisão: 26/04/2004

Abrangência

Versão 7.10

Versão 8.11

Definição

Um método de um WebService consiste em uma ação a ser disponibilizada no serviço. Damos a ela um nome para identificação, declaramos a mesma na estrutura da classe do Serviço, bem como seus parâmetros e respectivo retorno.

Parâmetros

Ao declarar o fonte de um método, o mesmo pode receber um ou mais parâmetros, de tipo básico e/ou estruturas, e inclusive pode não receber parâmetro algum. Neste caso, devemos especificar que o parâmetro recebido será NULLPARAM, ou seja, nenhum parâmetro.

Retorno

Um método de WebServices deve obrigatoriamente ter uma propriedade de retorno. Não faz parte da especificação de WebServices a criação de um método que não possua retorno.

Codificando o método em Advpl

Como visto anteriormente, tanto os parâmetros quanto o retorno de um método de WebServices deve ser declarado como um dado da classe (através da instrução WSDATA).

Quando escrevemos um método de um WebService, e o mesmo recebe uma solicitação de processamento, as propriedades declaradas como parâmetros do método são alimentadas, e o método é executado. Por tratarem-se de propriedades, o código fonte Advpl deverá interagir com estas propriedades, prefixando-as com '::' dois pontos seguidos), ou 'self:', sendo isto válido tanto para os parâmetros do método, como para a propriedade de retorno.

Dada a existência de uma LIB de Infra-Estrutura, que realiza a camada de comunicação, validação, montagem e desmontagem de pacotes; ao codificar um método de WebService existem sempre dois retornos : A propriedade de retorno do método, e o retorno efetivo do método ao final do processamento.

O retorno efetivo do método deve ser um valor booleano : Se retornado .T. (*True*) , isto indica à LIB, que o método foi executado com sucesso, e consequentemente a propriedade de retorno foi alimentada. Logo, o pacote 'SOAP' de retorno do método será montado pela LIB, e devolvida automaticamente ao 'Client' que solicitou a chamada de processamento.

Caso o retorno efetivo do método seja .F. (*False*), isto indica à LIB que, por alguma razão tratada no fonte do método, não foi possível a execução do método. Neste caso, devemos especificar, antes do retorno, através da função SetSoapFault(), a causa da impossibilidade de processamento.

Exemplo

```
WSMETHOD GetDate WSRECEIVE NULLPARAM WSEND Horario
WSSERVICE ServerTime
```

```
If dow(date())=1
    // Seta um soap_fault, informando que este serviço não é disponível aos domingos
    SetSoapFault('Metodo não disponível','Este serviço não funciona aos Domingos.')
    // e retorna .F., indicando que o serviço não foi processado com sucesso.
    Return .f.
Endif

// alimenta a propriedade de retorno
::Horario := time()
// E retorna .T. indicando processamento do método com sucesso
Return .T.
```

Atenção :

- Sempre que o retorno efetivo do método é verdadeiro (.T.), a propriedade de retorno deve ser preenchida. Caso ela não seja preenchida, a LIB irá retornar ao client solicitante um pacote de SOAP Fault, indicando que houve um erro no processamento do serviço, e registrar um error.log na estação servidora. Será gerado também uma ocorrência de erro, caso o método retorne .T., porém a função SetSoapFault() tenha sido chamada durante a execução do método. A ocorrência gerada é **<SERVICO> : <METODO> RETURN .T. WITH SOAP FAULT EXCEPTION NOT EMPTY.**
- Sempre que o retorno efetivo do método é falso (.F.), a função SetSoapFault() deve ser chamada, para que a LIB gere um pacote com o motivo do erro para o 'Client' que solicitou o método. Caso o retorno efetivo seja .F. , e a função SetSoapFault() não tenha sido chamada, é devolvido à estação 'Client' solicitante do processamento um Soap:Fault , com a ocorrência de erro **<SERVICO> : <METODO> RETURN .F. WITH SOAP FAULT EXCEPTION EMPTY.**

07. Tratamento de Erro dos WebServices

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

Dada a infra-estrutura envolvida no processamento dos WebServices, a rotina de tratamento de erro da aplicação WebServices 'Server' prevê o tratamento de ocorrências, desde advertência de carga dos serviços, até falhas de inicialização de ambiente, passando por erros que invalidam um determinado serviço compilado, até as ocorrências de inconsistências de parâmetros de chamada do serviço, inconsistências de retorno, ocorrências de erro fatal de processamento na aplicação, e ocorrências de processamento que não constituam um erro fatal, porém devem retornar um pacote de ocorrência de erro, conhecido por *SOAP FAULT*.

Os tratamentos aplicados às ocorrências reproduzidas no momento da carga do ambiente de WebServices estão relacionados no tópico "Falhas de Carga dos Serviços", os relacionados à ocorrências de erro fatal de execução dos serviços estão em "Ocorrências de Erro Fatal", e a discriminação da utilização do Soap Fault está descrita em "Utilização do SOAP FAULT".

08. Utilização do SOAP FAULT

Revisão: 30/04/2004

Abrangência

Versão 7.10

Versão 8.11

Quando desenvolvemos um serviço, e temos a necessidade de retornar ao 'Client' solicitante do processamento, uma ocorrência de falha não-fatal de um determinado processamento, deve ser retornado ao mesmo um pacote SOAP, que indica a causa da falha. Este pacote é conhecido por '*SOAP FAULT*'. A rotina de tratamento de erro fatal de execução do WebService, quando da ocorrência de tal, gera automaticamente um '*SOAP FAULT*' com a descrição resumida da ocorrência ao client solicitante.

Dado que, a camada da lib, responsável pela interpretação do pacote SOAP recebido pelo serviço, já se encarrega de validar o formato do pacote e conteúdos obrigatórios, um Web Service escrito em Advpl deve, antes de realizar o processamento proposto, validar se o conteúdo dos parâmetros está dentro da faixa de dados esperada, e condizentes com o esperado; para então realizar o processamento e retornar ao client solicitante.

Para inserir as excessões de execução com Soap-Fault, em um serviço 'Server', utilizamos a função SetSoapFault().

Soap-Faults padrão do Servidor Protheus de WebServices

A camada de comunicação da infra-estrututa de WebServices, realiza automaticamente os tratamentos de protocolo, formato do pacote SOAP e parâmetros obrigatórios. Caso exista alguma inconsistência na chamada de um serviço, que incorra em alguma destas excessões, o serviço solicitado não é chamado, e o servidor Protheus devolve automaticamente ao client solicitante um *Soap-Fault*, indicando o que aconteceu.

Estas ocorrências de *Soap-Fault* são mostradas no console do servidor Protheus, e são armazenadas também no arquivo error.log do ambiente utilizado.

Soap-Faults padrão após processamento do serviço

A camada de comunicação da infra-estrututa de WebServices valida também a montagem do pacote de retorno. Caso exista alguma propriedade de retorno obrigatório do serviço que não esteja alimentada de forma correta, o servidor Protheus devolve automaticamente ao client solicitante um *Soap-Fault*, indicando que ocorreu um erro interno no servidor de WebServices.

09. Serviço de Exemplo : SERVERTIME

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

Inicialmente, o exemplo proposto têm o objetivo de montar um WebService que retorne o horário no servidor Protheus. Para tal, será criado um serviço, com apenas (inicialmente) um método. A este serviço, daremos a ele o nome de SERVERTIME. E, ao método de buscar o horário no servidor, daremos o nome de GETSERVERTIME.

A operação de buscar o horário atual no servidor não necessita de nenhum parâmetro para a execução. Porém, ela terá um retorno : O horário atual , no formato 'hh:mm:ss'. A especificação de um WebServices permite que um serviço seja declarado de modo a não receber nenhum parâmetro, porém exige que o WebService sempre possua um retorno.

Codificando o Serviço

Para codificar um serviço, devemos utilizar o Protheus IDE, e criar um novo arquivo de programa, e nele escrever o serviço. A numeração disposta à esquerda do código-fonte é meramente ilustrativa, não devendo ser digitada. Ela é utilizada mais abaixo, onde este código é detalhado linha a linha.

```
1      #INCLUDE 'PROTHEUS.CH'
2      #INCLUDE 'APWEBSRV.CH'
3
4      WSSERVICE SERVERTIME
5          WSDATA Horario          as String
6          WSMETHOD GetServerTime
7      ENDWSSERVICE
8
9      WSMETHOD GetServerTime WSRECEIVE NULLPARAM WSEND Horario
WSSERVICE SERVERINFO
10     ::Horario := TIME()
11     Return .T.
```

- Linha 1 É especificada a utilização do Include “Protheus.ch”, contendo as definições dos comandos ADVPL e demais constantes
- Linha 2 Também especificamos a o Include “ApWebSrv.ch”, que contém as definições de comandos e constantes utilizados nas declarações de estruturas e métodos dos *Web Services*. Ele é obrigatório para o desenvolvimento de WebServices.
- Linha 4 Com esta instrução, é definido o inicio da classe do serviço principal, ao qual demos o nome de SERVERTIME
- Linha 5 Dentro da estrutura deste serviço, é informado que um dos parametros utilizados chama-se horário, e será do tipo String

Linha 6 Dentro da estrutura deste serviço, é informado que um dos métodos do serviço chama-se `GetServerTime` .

Linha 7 Como não são necessárias mais propriedades ou metodos neste serviço, a estrutura do serviço é fechada com esta instrução..

Linha 9 Aqui é declarado o fonte do Método `GetServerTime`, que não receberá parametro nenhum (mas para efeitos de declaração deve ser informado que ele receberá o parametro `NULLPARAM`), e é informado que seu retorno será o dado Horario (declarado na classe do serviço como uma propriedade, do tipo `String`) .

Linha 10 É atribuído na propriedade `::Horario` da classe deste serviço, o retorno da função `Advpl Time()`, que retorna a hora atual no servidor no formato `HH:MM:SS`. Devemos utilizar o `::`, para alimentarmos a propriedade da classe atual

Linha 11 O método `GetServerTime` é finalizado nesta linha, retornando `.T. (true)`, indicando que o serviço foi executado com sucesso.

Após compilado o serviço, deve ser acessada novamente a página de índice de serviços (`wsindex.apw`), e verificar se o novo serviço compilado lá se encontra.

Testando o Serviço

Ao acessar a página de índice, e constatarmos a existência do serviço, devemos obter o link através do qual o WSDL deste serviço está sendo fornecido, e utilizarmos de uma ferramenta para gerar um 'Client' que possibilite o uso deste serviço. É possível, inclusive, utilizar o Protheus IDE para gerar o fonte 'Client' para testar o serviço; porém existe a necessidade de criar uma função para instanciar a classe 'Client' gerada, alimentar os parâmetros e testar o serviço.

A partir da versão Protheus 8, podemos apenas gerar um fonte 'Client' desta classe, e compilá-lo no mesmo repositório do ambiente utilizado pelo WebServices 'Server', que a própria interface de Índice de Serviços irá permitir o teste do mesmo.

Falhas de Carga dos Serviços

Revisão: 06/05/2004

Abrangência

Versão 7.10

Versão 8.11

Neste tópico são abordadas as mensagens de ocorrências relacionadas à carga dos serviços.

Durante a inicialização do *engine* de Web Services, os serviços compilados são validados, e um ambiente é montado por *thread* para o atendimento de solicitações de processamento. Neste processo, existem ocorrências, relacionadas à montagem do ambiente, que podem impossibilitar a operação dos WebServices como um todo; e ocorrências que podem invalidar apenas um serviço, em caso de inconsistência da declaração do mesmo.

Erro de Estrutura : ARRAY OF em parametro de en...

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

XXX : Erro de Estrutura : ARRAY OF em parametro de entrada direto nao suportado.

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando um parâmetro [XXX] foi utilizado como parâmetro de entrada direto de um WebService, porém o mesmo foi declarado com tratamento de 'Array Of'. Não é suportado receber diretamente um array como parâmetro de um método de WebServices 'Server'.

Verifique e corrija o código-fonte, e crie uma estrutura intermediária para encalsular o parâmetro que deve ter tratamento de Array.

Erro de Estrutura : Estrutura Indefinida.

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

[XXX] : Erro de Estrutura : Estrutura Indefinida.

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando uma propriedade da classe server foi especificado como sendo uma estrutura (tipo não-básico), porém a declaração da estrutura não foi localizada.

Verifique e corrija o código-fonte e proceda com a declaração da referida estrutura.

Erro de Estrutura : Nome de Estrutura Inválido ...

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

[XXX] Erro de Estrutura : Nome de Estrutura Inválido - Tipo básico conflitante.

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando o nome de uma determinada estrutura [XXX] foi especificado com um nome igual a um tipo básico de informação. Esta ocorrência invalida apenas o serviço que utiliza a determinada estrutura.

Verifique e corrija o código-fonte e a declaração do tipo da estrutura.

Erro de Método : Estrutura de Entrada não encontrada...

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

[XXX] : [YYY] : Erro de Método : Estrutura de Entrada não encontrada.

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando um determinado método [XXX] foi especificado com algum parâmetro de entrada [YYY], cuja declaração não foi encontrada como uma propriedade no fonte construtor do serviço.

Verifique e corrija o código-fonte, e declare o parâmetro YYY como uma propriedade da classe XXX

Erro de Método : Estrutura de Retorno não encon...

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

[XXX] : [YYY] : Erro de Método : Estrutura de Retorno não encontrada.

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando um determinado método [XXX] foi especificado com uma estrutura [YYY], cuja declaração não foi encontrada como uma propriedade no fonte construtor do serviço.

Verifique e corrija o código-fonte, e declare a propriedade YYY como uma propriedade da classe XXX

Erro de Método : Estrutura de Retorno não pode ...

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

[XXX] : [YYY] : Erro de Método : Estrutura de Retorno não pode ser recebida como parâmetro.

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando um determinado método [XXX] foi declarado para receber uma estrutura [YYY] e retornar a mesma estrutura [YYY] . Isto não é suportado pelos WebServices do Protheus.

Verifique e corrija o código-fonte.

Erro de Método : Método [XXX] do Serviço [YYY] ...

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

Erro de Método : Método [XXX] do Serviço [YYY] não declarado no Serviço.

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando um determinado método [XXX], referente ao serviço [YYY], foi codificado, porém não foi declarado no construtor do Web Service. Esta ocorrência invalida apenas o serviço que utiliza a determinada estrutura.

Verifique e corrija o código-fonte e proceda com a declaração do método no construtor do serviço.

Erro de Método : Nome de Método Inválido - Tipo...

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

[XXX] Erro de Método : Nome de Método Inválido - Tipo básico conflitante.

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando o nome de uma determinada método [XXX] foi especificado com um nome igual a um tipo básico de informação. Esta ocorrência invalida apenas o serviço que utiliza o determinado método.

Verifique e corrija o código-fonte e a declaração do nome do método.

Erro de Estrutura : Redundancia de Estruturas

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

Durante a etapa de validação dos serviços, na carga dos WebServices, esta ocorrência é reproduzida quando temos uma cadeia de estruturas, compostas por tipos básicos e outras estruturas, e a declaração das estruturas entre em redundância. Por exemplo, declaramos a estrutura <A>, que tem dentro dela uma outra propriedade que é do tipo <A>, ou a estrutura <A> têm uma propriedade de tipo , e por sua vez tem uma propriedade do Tipo <A>.

Verifique e corrija o código-fonte e corrija a declaração das estruturas envolvidas.

WSDL Server ONLOAD ERROR - Falha Interna na ...

Revisão: 27/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSDL Server ONLOAD ERROR - Falha Interna na Carga do WebService

Esta ocorrência é apresentada na tela de índice dos WebServices (wsindex.apw), quando algum erro fatal ocorra na carga dos WebServices. Os detalhes sobre a ocorrência fatal são mostrados no console do servidor Protheus, e gravados no arquivo error.log do ambiente em uso.

Ocorrências de Erro Fatal - AUTOMATIC URLLOCATION FAILED

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Ao configurarmos um WebService 'Server', devemos especificar, através da chave URLLOCATION , a url específica para o acesso aos serviços.

Quando não definimos esta URL, a lib de WebServices identifica automaticamente sob qual *host* o serviço foi acessado. Esta operação não é possível quando o *header HTTP* do pacote informe uma operação diferente de "GET" ou "POST", ou o servidor Protheus está sendo executado em sua versão ISAPI, em conjunto com o Microsoft (R) Information Service.

Caso não seja possível identificar o *host* sob o qual a chamada foi realizada, o WebService não é processado, e o processamento é abortado com a ocorrência de erro acima.

Ocorrências de Erro Fatal - BUILD [XXX] USING WEBSERVICES HTTPS NOT SUPPORTED

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Quando da carga inicial dos WebServices 'Server', a configuração URLLOCATION é criticada pela Lib. Caso seja especificado que o acesso será realizado via '*HTTPS*', e o build atual do servidor Protheus utilizado ainda não suporta a utilização do WebService sob o protocolo *HTTPS*.

INVALID URLLOCATION [XXX] ON [YYY]

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Quando da configuração do servidor Protheus para WebServices, caso especificada a configuração URLLOCATION, porém a mesma não seja especificada com uma sintaxe válida, o processamento é abortado na subida das *Working Threads* do servidor, com esta ocorrência de erro fatal, indicando em [XXX] a url informada, e em [YYY] o nome do arquivo de configuração do servidor Protheus.

Uma url é considerada inválida, caso ela não seja iniciada com 'http://' ou 'https://', seja finalizada com um caractere não-alfanumérico ou diferente de '/', ou possua caracteres acentuados ou espaços. São considerados válidos apenas caracteres alfanuméricos, e os caracteres ':' (dois pontos), '.' (ponto), '/' (barra) e '-' (hífen).

Esta validação foi implementada na Infra-Estrutura de Web Services a partir da versão de infra-estrutura 'ADVPL WSDL Server 1.031209'

Ocorrências de Erro Fatal - **REQUIRED** **Return property [X] AS ARRAY OF [Y]** **IS ..**

Revisão: 23/04/2004

Abrangência

Versão 7.10

Versão 8.11

REQUIRED Return property [X] AS ARRAY OF [Y] IS EMPTY

Esta ocorrência de erro, é reproduzida quando do término do processamento de um método de um WebServices, na camada da LIB, quando da geração do pacote 'SOAP' de retorno ao 'Client' solicitante do serviço.

Quando da identificação da propriedade [X] de retorno obrigatório do método , a mesma deveria ser um 'Array' Advpl, contendo no mínimo um elemento; porém o array não continha nenhum elemento.

Verifique o método solicitado, e certifique-se que a propriedade de retorno esteja sendo alimentada.

Ocorrências de Erro Fatal - **REQUIRED** Return property [X] Type [Y] Unexpected

...

Revisão: 23/04/2004

Abrangência

Versão 7.10

Versão 8.11

REQUIRED Return property [X] Type [Y] Unexpected Valtype [Z]

Esta ocorrência de erro, é reproduzida quando do término do processamento de um método de um WebServices, na camada da LIB, quando da geração do pacote 'SOAP' de retorno ao 'Client' solicitante do serviço.

Quando da identificação da propriedade **obrigatória** [X] de retorno do método , a mesma deveria ser alimentada com um conteúdo Advpl do tipo [Y], porém, ao invés deste, ela continha um valor do tipo Advpl [Z].

Verifique o método solicitado, e certifique-se que a propriedade de retorno esteja alimentada com um conteúdo do tipo [Y], em conformidade com a declaração da propriedade no serviço.

Ocorrências de Erro Fatal - Return property [X] AS ARRAY Type [Y] Unexpected..

Revisão: 23/04/2004

Abrangência

Versão 7.10

Versão 8.11

Return property [X] AS ARRAY Type [Y] Unexpected Valtype [Z]

Esta ocorrência de erro, é reproduzida quando do término do processamento de um método de um WebServices, na camada da LIB, quando da geração do pacote 'SOAP' de retorno ao 'Client' solicitante do serviço.

Quando da identificação da propriedade [X] de retorno do método , a mesma deveria ser um 'Array' Advpl, contendo elementos do tipo [Y], porém, ao invés da propriedade ser um do Tipo A (Array), ela continha um valor do tipo Advpl [Z].

Verifique o método solicitado, e certifique-se que a propriedade de retorno esteja alimentada com um array,

Ocorrências de Erro Fatal - Return property [X] AS OBJECT Type [Y] Unexpected ..

Revisão: 23/04/2004

Abrangência

Versão 7.10

Versão 8.11

Return property [X] AS OBJECT Type [Y] Unexpected Valtype [Z]

Esta ocorrência de erro, é reproduzida quando do término do processamento de um método de um WebServices, na camada da LIB, quando da geração do pacote 'SOAP' de retorno ao 'Client' solicitante do serviço.

Quando da identificação da propriedade [X] de retorno do método , a mesma deveria ser uma Estrutura (Tipo Advpl 'O' - Objeto) Advpl, do tipo [Y], porém a propriedade de retorno continha um valor do tipo Advpl [Z].

Verifique o método solicitado, e certifique-se que a propriedade de retorno seja alimentada com a respectiva estrutura, em conformidade com a declaração da propriedade da classe do serviço.

Ocorrências de Erro Fatal - Return property [X] Type [Y] Unexpected Valtype ..

Revisão: 23/04/2004

Abrangência

Versão 7.10

Versão 8.11

Return property [X] Type [Y] Unexpected Valtype [Z]

Esta ocorrência de erro, é reproduzida quando do término do processamento de um método de um WebServices, na camada da LIB, quando da geração do pacote 'SOAP' de retorno ao 'Client' solicitante do serviço.

Quando da identificação da propriedade [X] de retorno do método , a mesma deveria ser alimentada com um conteúdo Advpl do tipo [Y], porém, ao invés deste, ela continha um valor do tipo Advpl [Z].

Verifique o método solicitado, e certifique-se que a propriedade de retorno esteja alimentada com um conteúdo do tipo [Y], em conformidade com a declaração da propriedade no serviço.

Ocorrências de Erro Fatal - UNKNOWN ERROR : EMPTY HTTP RETURN

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

Quando do processamento de uma requisição de um método de WebServices 'Server', são executadas consistências de pré-processamento e pós-processamento. Todas as consistências internas realizadas têm uma mensagem de retorno. Quando do final da execução do serviço, independentemente de ocorrer um processamento com sucesso ou com falha (*SoapFault*), é verificado se o tratamento efetuado gerou um pacote com a mensagem de retorno.

Caso esta ocorrência seja reproduzida, ela indica que ocorreu uma falha não tratada, ou uma impossibilidade de geração do pacote de retorno. Até o momento, esta ocorrência não foi reproduzida sob nenhuma condição.

Ocorrências de Erro Fatal - [SVC] : [METHOD] as [X] : Tipo Inesperado de Ret..

Revisão: 06/05/2004

Abrangência

Versão 7.10

Versão 8.11

[SVC] : [METHOD] as [X] : Tipo Inesperado de Retorno do Método.

A ocorrência de erro acima é reproduzida, quando do término da execução de um método de uma classe 'Server' de WebServices. A LIB espera um valor booleano (.T. ou .F.) de retorno efetivo do método. Caso o retorno efetivo não seja booleano, o processamento é abortado com a ocorrência acima, identificando o serviço chamado em [SVC], o método em [METHOD], e o tipo do retorno efetivo retornado em [X].

Verifique o código-fonte do método do serviço, e certifique-se que o retorno efetivo do método seja sempre .T. ou .F.

Aplicações Protheus 'Client' de WebServices

Revisão: 30/04/2004

Abrangência

Versão 7.10

Versão 8.11

Definição de Client

Quando um Web Service 'Server' é criado e disponibilizado, junto dele também é disponibilizada a definição do serviço, seus argumentos, estruturas e retornos (WSDL) . Para a utilização de um Web Service, é necessário montar um programa –‘client’, que seja capaz de montar um “envelope” SOAP com os dados necessários ao processamento do Serviço, realizar a chamada, e tratar o pacote de retorno do serviço e suas respectivas exceções.

Embora existam Web Services que podem ser acessados via Http “direto”, apenas passando parâmetros via URL, o ‘client’ de Web Services do Protheus têm seu foco e recursos direcionados apenas a serviços que possuam interface de comunicação que realize POST de pacotes de dados XML em formato SOAP. O Protheus possui ferramentas e infra-estrutura incorporadas que permitem esta integração.

Geração do Client em Advpl

Utilizando o IDE, encontra-se disponível, no menu 'Ferramentas', a opção para que, através de um link para a obtenção do documento WSDL de um serviço, o Protheus gere automaticamente, em Advpl, uma classe 'Client' para a comunicação e utilização do mesmo.

Para tal, basta obtermos o endereço internet (*URL*) do WSDL desejado, criar um novo arquivo-fonte, e acessar o menu 'Ferramentas -> Gerar Cliente WebServices...'. Para cada serviço que se tenha a necessidade de geração de um fonte client, recomenda-se fortemente que cada fonte client seja gerado em um arquivo independente e exclusivo para este fim, e que de forma alguma este fonte gerado pelo assistente seja alterado.

Requisitos básicos para a Geração do Client em Advpl

O processo de geração de fonte é disparado através do IDE, porém é o servidor Protheus que irá buscar o documento WSDL solicitado. De modo que, a estação servidora utilizada no ambiente deve ter acesso ao endereço solicitado.

Aplicações Protheus 'Client' de WebServices - Geração de Client em Advpl - Passo 01

Revisão: 30/04/2004

Abrangência

Versão 7.10

Versão 8.11

Passo 1 : Determinar como obter o WSDL do serviço desejado

A maioria das definições WSDL dos serviços disponíveis na WEB são acessados através de uma URL, em geral apontando para o servidor onde o serviço está publicado, contendo o nome do serviço na url e um sufixo ?WSDL ou .WSDL na Url. Não há padrão definido para tal, de modo que cada servidor pode disponibilizar (ou não) o WSDL de uma maneira diferente . O WSDL de alguns serviços restritos (como por exemplo o serviço de busca na base de dados do Google) são disponibilizados em arquivo ASCII, enviados por e-mail, após um cadastro no site e autorização da empresa para o uso do serviço por ele provido.

No nosso exemplo ilustrativo, a definição do serviço é obtida diretamente via http, através do link <http://localhost/SERVETIME.apw?WSDL>. Caso este link seja acessado através de um Web Browser (Internet .Explorer., por exemplo), será exibido no browse um documento XML correspondendo a definição do serviço.

Aplicações Protheus 'Client' de WebServices

- Geração de Client em Advpl - Passo 02

Revisão: 30/04/2004

Abrangência

Versão 7.10

Versão 8.11

Passo 2 : Gerar o Fonte AdvPl do '*client*' usando o Assistente do IDE

Ao ser gerado um fonte '*client*' para um *Web Service*, este fonte conterá as definições dos metodos do serviço, a(s) estrutura(s) utilizada(s) no mesmo, e a(s) classe(s) intermediária(s) de uso interno para montagem e desmontagem da(s) estrutura(s) ; visando o encapsulamento de todos os tratamentos de envio e recebimento de dados através de pacotes SOAP.

O Fonte gerado através do assistente de criação de fonte deve preferencialmente ser gerado e compilado em um arquivo exclusivo, destinado unica e exclusivamente a este código. E, por tratar-se de uma classe Advpl gerada a partir da definição de um serviço, não deve ser inserida e/ou alterada nenhuma das definições geradas pelo assistente, pois as mesmas serão perdidas caso o fonte seja gerado novamente .

Para geração do fonte '*client*' em Advpl para utilizar este serviço, é necessário criar um novo arquivo .PRX no IDE, especificamente para conter as classes deste serviço . Então, deve ser acessado o menu "Ferramentas", opção "Gerar Ciente Webservices" . Neste momento, será mostrado na tela um pop-up semelhante ao mostrado abaixo :



No campo de entrada de dados, deve ser digitada a URL de onde o servidor irá obter a definição do Webservice. (no nosso caso, <http://localhost/SERVERTIME.apw?WSDL>) . Após a confirmação da janela acima, caso o processamento ocorra com sucesso, na janela de mensagens do Ide será mostrado um texto semelhante ao abaixo :

Estabelecendo conexão com o server...

Por favor aguarde. Obtendo descrição do Webservice...

Finalizando conexão com o server...

Ok

E, na janela do novo arquivo criado, deverá ser criado um código-fonte semelhante ao mostrado abaixo :

```
#INCLUDE 'PROTHEUS.CH'
#INCLUDE 'APWEBSRV.CH'

--- header do serviço ---

/*
=====
=====
WSDL Location   http://localhost/SERVERTIME.apw?WSDL
Gerado em      12/30/02 17:21:29
Observações    Código-Fonte gerado por ADVPL WSDL Client 1.021217 B
                Alterações neste arquivo podem causar funcionamento incorreto
                e serão perdidas caso o código-fonte seja gerado novamente.
=====
===== */

/* -----
WSDL Service WSSERVERTIME
----- */

--- declaração da Classe 'client' do WebService, com metodos e propriedades utilizadas ---
WSCLIENT WSSERVERTIME

    WSMETHOD NEW
    WSMETHOD GETSERVERTIME

    WSDATA _URL          AS String
    WSDATA cGETSERVERTIMERESULT  AS string

ENDWSCLIENT

--- declaração do método NEW, para a criação do Objeto / Serviço ---
WSMETHOD NEW WSCLIENT WSSERVERTIME
    ::_URL          := NIL
    ::cGETSERVERTIMERESULT := "
Return Self

/* -----
WSDL Method GETSERVERTIME of Service WSSERVERTIME
----- */

--- Definição do método, que recebe os parâmetros de chamada, executa o serviço e alimenta as propriedades de retorno do metodo, contendo os encapsulamentos necessários para tratamento de excessões ---

WSMETHOD GETSERVERTIME WSEND NULLPARAM WSRECEIVE cGETSERVERTIMERESULT WSCLIENT WSSERVERTIME
Local cSoap := "", oXmlRet

BEGIN WSMETHOD

DEFAULT ::_URL := 'http://localhost/SERVERTIME.apw'

cSoap += '<GETSERVERTIME xmlns='http://localhost/'>'
cSoap += '</GETSERVERTIME>'

oXmlRet := SvcSoapCall( Self,cSoap,;
    'http://localhost/GETSERVERTIME';;
    'DOCUMENT','http://localhost/',)

```

```
::cGETSERVERTIMERESULT := xGetInfo( oXmlRet,  
'_GETSERVERTIMERESPONSE:_GETSERVERTIMERESULT:TEXT', " )
```

```
END WSMETHOD
```

```
oXmlRet := NIL  
Return .T.
```

O fonte acima constitui uma Classe em Advpl, gerada para realizar a interface com a classe original publicada no Server, já realizando os tratamentos adequados para realizar a comunicação via http com o servidor onde o serviço está publicado. Vale observar que, as linhas em negrito no fonte acima não foram inseridas pelo assistente do IDE, mas acrescentadas a este documento para fins didáticos.

O cabeçalho do fonte contém informações sobre a localização do WSDL utilizado para a geração do fonte, data e hora de geração e versão do engine de *Web Services* utilizado . Logo abaixo, a declaração de uma classe '*client*' de *Web Services* (WSCIENT WSSERVERTIME), com o método new() para inicialização das propriedades advpl da classe . E, em seguida, a declaração do método de busca de Horário (WSMETHOD GETSERVERTIME), que não envia parâmetro algum, e retorna o horário atual do server em :: cGETSERVERTIMERESULT, com todos os tratamentos necessários embutidos.

Aplicações Protheus 'Client' de WebServices

- Geração de Client em Advpl - Passo 03

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Passo 3 : Criar um fonte que utilize esta classe para utilização do WebService.

Agora, é necessário criar um novo arquivo no IDE, e montar uma função para utilizar a classe de *Web Services 'client'* para obter o horário no servidor.

```
1      #INCLUDE 'PROTHEUS.CH'
2
3      User Function TestClient()
4      Local oSvc := NIL
5
6      oSvc := WSSERVERTIME():New()
7
8      If oSvc:GETSERVERTIME()
9          alert('Horário no Servidor : '+ oSvc:cGETSERVERTIMERESULT)
10     Else
11         alert('Erro de Execução : '+GetWSCError())
12     Endif
13
14     Return
```

Linha 1	É declarada a utilização do Include “Protheus.ch”, contendo as definições dos comandos ADVPL e demais constantes
Linha 3	Inicia-se a definição da User Function para utilizar o <i>Web Service</i>
Linha 4	Uma variável local é declarada para conter o Objeto do <i>Web Service 'client'</i>
Linha 6	Utilizando-se do serviço, a variável oSvc é alimentada com uma nova instância do <i>Web Services 'client'</i> , obtida através da sintaxe <NOME_DO_SERVICO>():New()
Linha 8	É executado o método GetServerTime a partir do Objeto do serviço oSrv, sem passar qualquer parametro. O retorno de um método do <i>'client'</i> pode ser .T. (true) em caso de execução com sucesso ou .F. (false) em caso de falha de execução .
Linha 9	Caso o serviço tenha sido executado com sucesso, o retorno esperado é alimentado na propriedade cGetServerTimeResult do objeto do serviço.
Linha 11	Caso contrário (retorno .F.), ocorreu alguma falha na chamada do serviço, como por exemplo o servidor não estava no ar, demorou muito pra responder (time-out), entre outras. Para ser possível recuperar maiores detalhes sobre a ocorrência de erro, deve ser utilizada a função GetWSCError(), que retorna uma string com o resumo da ocorrência .
Linha 13	O programa de teste é finalizado com um Return

Aplicações Protheus 'Client' de WebServices - Geração de Client em Advpl - Passo 04

Revisão: 22/04/2004

Abrangência

Versão 7.10

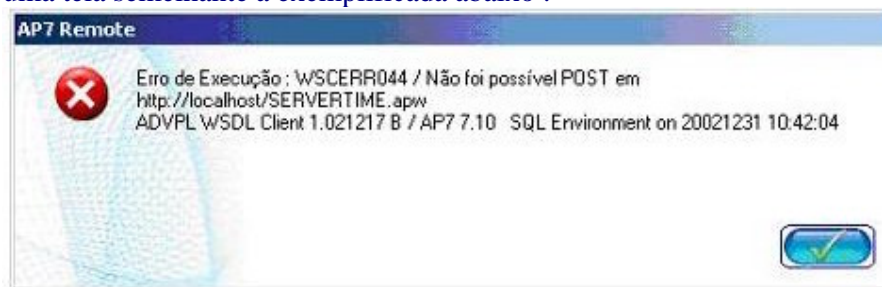
Versão 8.11

Passo 4 : Executar o programa de testes

Abra uma nova instância do Ap Remote, e execute a função U_TESTCLIENT .
Caso o *Web Service* esteja no ar e funcionando, e o fonte '*client*' seja devidamente compilado e sem erros, o resultado esperado é uma janela semelhante a mostrada abaixo:



No ambiente montado para teste, o Servidor de *Web Services* e o '*client*' estão no Protheus, compilados no mesmo Repositório de Objetos . Para fins didáticos, é possível simular uma ocorrência de falha no '*client*', ao desabilitar o Server HTTP do Protheus (colocando enable=0 na chave [http] do arquivo de consiguração do servidor), re-iniciar o Server Protheus, e executar o programa '*client*' novamente . Deve ser obtida uma tela semelhante a exemplificada abaixo :



Aplicações Protheus 'Client' de WebServices - Geração de Client em Advpl - Passo 05

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Passo 5 : Obtendo informações de “debug”

Visto até o passo 4, um exemplo completo de um ‘client’ funcionando perfeitamente . Agora, é possível imaginar que, durante o desenvolvimento e testes do ‘client’ do serviço, façam-se necessárias determinadas informações internas as rotinas de execução do serviço no ‘client’ Advpl . Para tal, foi criada uma função que permite definir em tempo de execução, um nível de detalhamento de informações adicionais relacionadas ao *Web Service* ; informações estas que serão mostradas no Console do Server Protheus (caso habilitado) . a Função para definir o nível de detalhe chama-se WSDLDbgLevel(), e recebe um número como parâmetro :

0 (default)	Sem informações adicionais.
1	Apenas String SOAP de retorno do Server.
2	Strings Soap de Envio e Retorno.

Então, na linha 7, é acrescentada a instrução WSDLDbgLevel(2), para ativar o nível mais completo de informacoes adicionais, e é possível observar no console do servidor as mensagens apresentadas durante a execução do fonte de testes do ‘client’ . Deve ser obtido um echo no console do server semelhante ao exemplo abaixo :

Iniciando Thread (siga0984, AUTOMAN)...

```
-----
SvcSoapCall to http://automan:8000/webservice/SERVERTIME.apw /
DOCUMENT
Namespace http://automan:8000/webservice/
SoapAction http://automan:8000/webservice/GETSERVERTIME
Called from GETSERVERTIME ( 137)
Called from U_TESTCLIENT ( 10)
----- SOAPSEND -----
<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:xsd='
http://www.w3.org/2001/XMLSchema'
xmlns:soap='http://schemas.xmlsoap.org/soap/en
velope/'> <soap:Body>
```

```

<GETSERVERTIME xmlns='http://automan:8000/webservice/'>
</GETSERVERTIME> </soap:Body>
</soap:Envelope>
-----
----- POST RETURN -----
<?xml version='1.0' encoding='utf-8'?><soap:Envelope
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:so
ap='http://schemas.xmlsoap.org/soap/envelope/'><soap:Body><GETSERVERT
IMERESPONSE
xmlns='http://automan:8000/webservice/'><GETSERVERTIMERESULT>10:37:1
0</GETSERVE
RTIMERESULT></GETSERVERTIMERESPONSE></soap:Body></soap:Envelope
>
-----

```

Fim Thread (sigma0984, AUTOMAN) BytesIn 73 BytesOut 75

O texto **marcado em azul claro** são as mensagens informativas a respeito da chamada do Webservice, informando a URL chamada, o estilo soap de troca de dados (**document**), o Namespace e o SoapAction utilizados. O Texto **marcado em verde** (SOAPSEND) informa o conteúdo do pacote Soap que foi enviado (postado) ao Servidor, e o **conteúdo em amarelo** (POST RETURN) informa o conteúdo do pacote Soap devolvido pelo Server referente a esta solicitação.

Quando ocorre um erro qualquer, relacionado ‘a execução do ‘client’ Web Services, o método chamado retorna .F., e o erro pode ser recuperado através da função GetWSCError(), vista anteriormente . Para cada excessão prevista no ‘client’, existe um código de erro correspondente, todos eles prefixados com WSCERR . A maioria das ocorrências está relacionada ‘a geração do Código fonte do ‘client’ Advpl utilizado-se o IDE. Todas as ocorrências de excessão tratadas pelo Web Services ‘client’ Advpl estão relacionadas no Tópico Web Services ‘client’ – Códigos de Erro .

Aplicações Protheus 'Client' de WebServices

- Tipos de dados suportados - 'Client'

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Até o momento, são suportadas as gerações de código Advpl para WebServices 'Client', que utilizam os tipos básicos de dados listados abaixo. Para permitir a manipulação de cada tipo, utilizando variáveis Advpl, são utilizados os tipos básicos do Advpl para tratar simultaneamente mais de um tipo de dado dos pacotes 'SOAP' dos WebServices.

Os tipos abaixo são disponibilizados em Advpl através de uma variável de tipo 'N' Numérica

- INT
- INTEGER
- BYTE
- FLOAT
- DOUBLE
- UNSIGNEDLONG
- UNSIGNEDINT
- DECIMAL
- LONG

Os tipo abaixo é disponibilizado em Advpl através de uma variável de tipo 'D' Data

- DATE

Os tipos abaixo são disponibilizados em Advpl através de uma variável de tipo 'C' Character

- STRING
- DATETIME
- CHAR (**)
- BASE64BINARY

(**) O tipo CHAR corresponde à uma string, contendo o número do caractere correspondente à tabela ASCII

Os tipo abaixo é disponibilizado em Advpl através de uma variável de tipo 'L' Logica

- BOOLEAN

WSCERR000 / WSDL não suportado. Existe mais de ..

Revisão: 22/04/2004

Abrangência

Versão 8.11

[WSDL não suportado. Existe mais de um serviço declarado.]

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Por definição, um WSDL deve conter um e apenas um serviço declarado, com um ou mais métodos . Caso sejam identificados mais de um serviço no mesmo WSDL, no momento da geração do fonte, o processo é abortado, o WSDL é considerado inválido, e o fonte client não é gerado.

WSCERR001 / Não há SOAP:BINDINGS para a geração ..

Revisão: 22/04/2004

Abrangência

Versão 8.11

WSCERR001 / Não há SOAP:BINDINGS para a geração do Serviço.

Durante a geração do código-fonte para 'client' Advpl, a partir de uma definição de serviço (WSDL), uma vez identificado o serviço, o gerador de código procura a declaração dos BINDINGS no WSDL. Caso esta declaração não esteja presente, a rotina considera o WSDL incompleto, e aborta o processo de geração de código com esta mensagem.

WSCERR003 / [XXX / YYY]

Enumeration não suportado

Revisão: 22/04/2004

Abrangência

Versão 8.11

WSCERR003 / [XXX / YYY] Enumeration não suportado

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço. Quando encontrada uma estrutura básica (*SimpleType*), onde foi especificado um '*enumeration*' (lista de parâmetros válidos pré-determinada), são suportados os seguintes tipos básicos de parâmetros, listados abaixo :

- STRING
- FLOAT
- DOUBLE
- DECIMAL
- INT
- INTEGER
- LONG
- UNSIGNEDINT
- UNSIGNEDLONG

Caso o WSDL contenha um '*enumeration*', utilizando um tipo de dado diferente dos declarados acima, o processo de geração de fonte é abortado com a ocorrência de erro acima, onde o '*enumeration*' não suportado é identificado em <XXX> e <YYY>, correspondendo ao nome do parâmetro e tipo utilizado, respectivamente.

WSCERR004 / NAO IMPLEMENTADO (001<X> / <N> / ...

Revisão: 22/04/2004

WSCERR004 / NAO IMPLEMENTADO (001<X> / <N> / WSDLTYPE_NAME)

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, uma estrutura contenha um determinado elemento, que aponte para uma outra estrutura, e esta não seja encontrada no WSDL (ocorrência <X> = A), ou seja encontrada - porém registrada não como uma estrutura (*complextype*)- (ocorrência <X> = B), o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a estrutura pendente em <WSDLTYPE_NAME>.

WSCERR006 / WSDL inválido ou não suportado.

Revisão: 22/04/2004

Abrangência

Versão 8.11

WSCERR006 / WSDL inválido ou não suportado.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, um parâmetro de primeiro nível (message) do WSDL for especificado sem nome, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima.

WSCERR007 / WSDL inválido ou não suportado.

Revisão: 22/04/2004

Abrangência

Versão 8.11

WSCERR007 / WSDL inválido ou não suportado.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, um parâmetro de primeiro nível (message) do WSDL for especificado sem definição de tipo, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima.

WSCERR008 / Retorno NULLPARAM inválido.

Revisão: 22/04/2004

Abrangência

Versão 8.11

WSCERR008 / Retorno NULLPARAM inválido.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, um parâmetro de retorno do WSDL seja identificado como 'retorno nulo', o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima.

WSCERR009 / INTERNAL ERROR (X)

Revisão: 29/04/2004

WSCERR009 / INTERNAL ERROR (X)

Esta é uma ocorrência de erro interna do 'engine' de geração de código-fonte Advpl, não reproduzida até o momento. Quando do processamento de um WSDL, os parâmetros e mensagens especificadas no WSDL são identificados internamente como parâmetros de entrada , parâmetro de saída , ou entrada e saída. Caso, após a análise inicial de parâmetros, algum parâmetro não seja enquadrado nestas definições, o processamento de geração é abortado com a ocorrência acima.

WSCERR010 / [STRUCT_TYPE] Estrutura / Tipo inc ...

Revisão: 22/04/2004

Abrangência

Versão 8.11

WSCERR010 / [STRUCT_TYPE] Estrutura / Tipo incompleto

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, caso uma estrutura complexa não contenha a especificação de seus elementos internos e a mesma não contenha nenhuma referência ao SCHEMA ou à outra estrutura, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, informando em [STRUCT_TYPE], o nome da estrutura incompleta.

WSCERR011 / Retorno NULLPARAM inválido.

Revisão: 22/04/2004

Abrangência

Versão 8.11

WSCERR011 / Retorno NULLPARAM inválido.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, um parâmetro de retorno do WSDL seja identificado como 'retorno nulo', o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima.

Observação : Esta ocorrência é semelhante à ocorrência WSCERR008, porém esta ocorrência (011) refere-se à uma sub-estrutura do serviço , e a primeira (008) refere-se à um parâmetro / estrutura de primeiro nível do serviço.

WSCERR012 / INTERNAL ERROR (X)

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR012 / INTERNAL ERROR (X)

Esta é uma ocorrência de erro interna do 'engine' de geração de código-fonte Advpl, não reproduzida até o momento. Quando do processamento de um WSDL, os parâmetros e mensagens especificadas no WSDL são identificados internamente como parâmetros de entrada, parâmetro de saída, ou entrada e saída. Caso, após a análise inicial de parâmetros, algum parâmetro não seja enquadrado nestas definições, o processamento de geração é abortado com a ocorrência acima.

Observação : Esta ocorrência é semelhante à WSCERR009, porem esta indica uma falha em outro ponto da rotina interna de análise.

WSCERR013 / [SOAP_TYPE] UNEXPECTED TYPE.

Revisão: 22/04/2004

WSCERR013 / [SOAP_TYPE] UNEXPECTED TYPE.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, um parâmetro de tipo básico não se encontre entre os tipos básicos suportados pelo engine 'Client' de WebServices do Protheus, a geração do fonte é abortada com esta ocorrência, indicando em SOAP_TYPE o tipo não suportado.

WSCERR014 / INVALID NULLPARAM INIT

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR014 / INVALID NULLPARAM INIT

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, para cada propriedade da estrutura do serviço são montadas as rotinas de inicialização de cada uma delas. Caso a rotina de geração de fonte receba a instrução de inicializar a propriedade reservada 'NULLPARAM', o processamento é abortado com esta ocorrência.

Esta ocorrência poderia ser causada por uma falha na validação inicial do WSDL, ou pela declaração de uma propriedade do tipo 'NULLPARAM'; e até o momento não foi reproduzida.

WSCERR015 / Node [XXX] as [YYY] on SOAP Resp ...

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR015 / Node [XXX] as [YYY] on SOAP Response not found.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, no momento que o client está desmontando o pacote SOAP retornado pelo serviço.

Caso o serviço utilize um *soap-style* RPC, e o node [XXX], correspondente ao retorno esperado do tipo [YYY] não for encontrado no pacote, o processamento do pacote de retorno é abortado com esta ocorrência.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR016 / Requisição HTTPS não suportada ...

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR016 / Requisição HTTPS não suportada neste Build. [XXX]

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Quando informada uma URL para buscar a definição do serviço (WSDL), utilizando o protocolo *HTTPS*; porém o Build do Protheus atual não suporta o tratamento de webservices em *HTTPS*, a geração do código-fonte é abortada com esta ocorrência de erro.

Para gerar um fonte 'Client' de WebServices, que utilize o protocolo *HTTPS*, o Build do Protheus deve ser atualizado.

WSCERR017 / HTTP[S] Retuição retornou [NIL]

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR017 / HTTP[S] Requisição retornou [NIL]

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Quando informada uma URL para buscar a definição do serviço (WSDL), utilizando o protocolo *HTTP ou HTTPS*; e não foi possível buscar o link solicitado, o processamento é abortado com a ocorrência acima.

Dentre as possíveis causas para esta ocorrência, podemos considerar :

- Sintaxe da URL inválida
- Servidor inválido, inexistente, ou DNF não disponível
- Servidor fora do ar

Verifique a URL digitada, e realize a requisição da mesma através de um Web Browser, para certificar-se que a mesma é válida e que a definição WSDL está realmente publicada e acessível sob o link informado.

WSCERR018 / HTTP[S] Requisição retornou [EMPTY]

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR018 / HTTP[S] Requisição retornou [EMPTY]

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Quando informada uma URL para buscar a definição do serviço (WSDL), utilizando o protocolo *HTTP ou HTTPS*; e não foi possível buscar o link solicitado, o processamento é abortado com a ocorrência acima.

Diferentemente da ocorrência WSCERR017, esta ocorrência foi reproduzida quando o servidor de WebServices que fornece o documento WSDL foi localizado, a requisição foi feita com sucesso, porém o servidor Protheus recebeu como retorno um pacote HTTP incompleto ou inválido.

Verifique a URL digitada, e realize a requisição da mesma através de um Web Browser, para certificar-se que a mesma é válida e que a definição WSDL está realmente publicada e acessível sob o link informado.

WSCERR019 / (XXX) Arquivo não encontrado.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR019 / (XXX) Arquivo não encontrado.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Quando informada uma URL para buscar a definição do serviço (WSDL), apontando para um arquivo no disco; porém o arquivo não foi encontrado, o processamento é abortado com a ocorrência acima.

Dentre as possíveis causas para esta ocorrência, podemos considerar :

- Diretório não existente ou inválido.
- Arquivo não existente ou inválido.
- Falta de permissão de acesso ao arquivo solicitado.

WSCERR020 / (XXX / FERROR YYY)

Falha de Abertura

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR020 / (XXX / FERROR YYY) Falha de Abertura.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Quando informada uma URL para buscar a definição do serviço (WSDL), apontando para um arquivo no disco; porém houve uma impossibilidade de acesso ao arquivo.

Dentre as possíveis causas para esta ocorrência, podemos considerar :

- Arquivo aberto em modo exclusivo por outra estação
- Falha de permissão / direito de abertura do arquivo

Verifique as propriedades e direitos do arquivo solicitado e repita a operação.

WSCERR021 / [INFO] WSDL Parsing [PARSER_WARNING]

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR021 / [INFO] WSDL Parsing [PARSER_WARNING]

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Quando informada uma URL para buscar a definição do serviço (WSDL), após o documento WSDL ser recuperado, caso seja detectada alguma inconsistência, considerada pelo parser interno de xml do Protheus como uma advertência (*warning*), no documento XML, o WSDL é considerado inválido e a geração do fonte é cancelada, com esta ocorrência. Em PARSER_WARNING é discriminada a mensagem de advertência do parser interno; e em [INFO] é especificado o documento / operação que apresentou a inconsistência.

WSCERR022 / [INFO] WSDL Parsing [PARSER_ERROR]

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR022 / [INFO] WSDL Parsing [PARSER_ERROR]

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Quando informada uma URL para buscar a definição do serviço (WSDL), após o documento WSDL ser recuperado, caso seja detectada alguma inconsistência, considerada pelo parser interno de xml do Protheus como erro no documento XML, o WSDL é considerado inválido e a geração do fonte é cancelada, com esta ocorrência. Em *[PARSER_ERROR]* é discriminada a ocorrência de erro do parser interno; e em *[INFO]* é especificado o documento / operação que apresentou a inconsistência.

WSCERR023 / [INFO] FALHA INESPERADA AO IMPORTAR ..

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR023 / [INFO] FALHA INESPERADA AO IMPORTAR WSDL

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. Quando informada uma URL para buscar a definição do serviço (WSDL), após o documento WSDL ser recuperado, caso o documento tenha passado pela etapa de validação do XML, onde o documento retornado constitui um XML sintaticamente válido, porém o parser não identifique nenhuma estrutura referente a um documento WSDL, o documento é considerado inválido, e a geração do fonte é cancelada, com esta ocorrência. Em [INFO] é especificado o documento / operação que apresentou a inconsistência.

WSCERR024 / [MSG_INFO] MESSAGE não encontrada.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR024 / [MSG_INFO] MESSAGE não encontrada.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, caso uma seção de mensagens (*message*) seja especificado para uma operação, porém não seja encontrado no WSDL, o mesmo é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a mensagem não encontrada em [MSG_INFO]. Caso a informação [MSG_INFO] estiver vazia, o documento WSDL não especificou alguma mensagem de parâmetro ou retorno na seção <portType> da lista de métodos do WSDL.

WSCERR025 / [BIND_INFO] Binding não Encontrado.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR025 / [BIND_INFO] Binding não Encontrado.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, caso uma seção de asmarração (*binding*) não seja localizado para uma operação especificada no WSDL, e a mesma não seja encontrada no WSDL, o mesmo é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a mensagem não encontrada em [BIND_INFO].

WSCERR026 / TARGETNAMESPACE não definido no WSDL.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR026 / TARGETNAMESPACE não definido no WSDL.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando é iniciado este processamento, é verificado se o documento WSDL contém a definição do *Namespace* de destino (*TargetNameSpace*) utilizado. Caso este não seja localizado, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima.

WSCERR027 / [OPER_INFO] BIND:OPERATION não enc ...

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR027 / [OPER_INFO] BIND:OPERATION não encontrado.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, caso uma operação / método do WebService não seja encontrada na seção de amarração (*binding*), o documento WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a operação não encontrada em [OPER_INFO].

WSCERR028 / [PORT_INFO] PortType não Encontrado ..

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR028 / [PORT_INFO] PortType não Encontrado em aPort.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, caso uma operação / método do WebService não seja encontrada na seção de portas do WSDL (*PortType*), o documento WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a porta não encontrada em [*PORT_INFO*].

WSCERR029 / [PORT_INFO] PortType não contém oper..

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR029 / [PORT_INFO] PortType não contém operações.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, caso uma operação / método do WebService não contenha a definição das operações na seção de portas do serviço (*PortType*), o documento WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a porta sem definição em [PORT_INFO].

WSCERR031 / [SCTUCT_NAME] Tipo sem NAMESPACE.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR031 / [SCTUCT_NAME] Tipo sem NAMESPACE.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando deste processamento, caso uma determinada estrutura seja identificada como sendo externa ao WSDL atual, referenciada por um *IMPORT* ou *REF*; se a estrutura estiver declarada no WSDL sem o referido namespace, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a estrutura incompleta em [STRUCT_NAME]

WSCERR032 / [SHORT_NS] NAMESPACE não encontrado.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR032 / [SHORT_NS] NAMESPACE não encontrado.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando do processamento de estruturas pendentes, identificadas como sendo externas ao WSDL atual, especificadas por um *IMPORT* ou *REF*, o namespace da mesma deve estar declarado no header do WSDL. Caso ele não seja encontrado, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando o namespace não encontrado em [SHORT_NS].

WSCERR033 / [LONG_NS] NameSpace sem Import decl ..

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR033 / [LONG_NS] NameSpace sem Import declarado

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Complementar ao erro **WSCERR032**, este é reproduzido quando o namespace identificado para o parâmetro seja externo ao WSDL, porém a URL para processamento do mesmo não seja especificada através de um *Import* no WSDL. Neste caso, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando o namespace não encontrado em *[LONG_NAMESPACE]*.

WSCERR034 / [INFO_NS] NAMESPACE sem LOCATION ...

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR034 / [INFO_NS] NAMESPACE sem LOCATION informado.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Complementar ao erro WSCERR033, este é reproduzido quando a declaração da URL / Location do *Namespace* externo não esteja declarado no *<IMPORT...>* do WSDL . Neste caso, o documento é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando o namespace incompleto em [INFO_NS] .

WSCERR035 / [TYPE] Tipo indefinido.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR035 / [TYPE] Tipo indefinido.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando do processamento de estruturas pendentes, identificadas como sendo externas ao WSDL atual, especificadas por um *IMPORT* ou *REF*, o namespace da mesma é identificado e importado, e todo o WSDL é re-processado. No reproprocessamento, caso o parâmetro / estrutura pendente não seja encontrado, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a estrutura pendente em *[TYPE]*

WSCERR036 / Definição não suportada.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR036 / Definição não suportada.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando da validação de estruturas complexas, caso a mesma não possua tipo definido, e não seja uma referência externa ao WSDL, ela deve ser uma referência ao próprio SCHEMA. Caso seja especificada qualquer outro tipo de referência, o WSDL não é suportado, e o processo de geração é abortado com a mensagem acima.

WSCERR037 / [TYPE] Estrutura Interna Inesperada.

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR037 / [TYPE] Estrutura Interna Inesperada.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando da validação de estruturas complexas, caso a mesma tenha passado por todas as interpretações cabíveis a uma estrutura, e mesmo assim não foi possível identificá-la, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando a estrutura em [TYPE].

WSCERR038 / [PARAM] WSDL inválido ou não suportado

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR038 / [PARAM] WSDL inválido ou não suportado.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE. No processo de geração, são analisados todos os parâmetros e estruturas utilizadas pelos métodos do serviço, até que todas as estruturas utilizadas sejam processadas.

Quando da validação de estruturas complexas, caso a mesma tenha passado por todas as interpretações cabíveis de uma estrutura, porém seu nome interno não foi declarado, o WSDL é considerado inválido, e o processo de geração é abortado com a mensagem acima, identificando o parâmetro de origem da mesma em [PARAM].

WSCERR039 / Unexpected DumpType [X]

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR039 / Unexpected DumpType [X]

Quando da utilização da função XMLDataSet, para a interpretação de um objeto de retorno XML em formato DataSet, caso não seja passado um objeto Advpl de tipo válido (Objeto XML ou Array), o processamento é abortado, mostrando a mensagem acima, identificando o tipo de parâmetro recebido em [X]

Verifique o código-fonte da aplicação e certifique-se de sempre passar um Objeto XML ou Array para a função XMLDataSet()

WSCERR040 / Unexpected SCHEMA Type [X]

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR040 / Unexpected SCHEMA Type [X]

Quando da utilização da função XMLDataSchema, para determinar os dados recebidos por um retorno de um Web Service que retorna uma referência ao Schema, e não seja passado a função um Objeto Advpl de Tipo Válido (Objeto Xml ou Array), o processamento é abortado, mostrando a mensagem acima, identificando o tipo de parâmetro recebido em [X]

Verifique o código-fonte da aplicação e certifique-se de sempre passar um Objeto XML ou Array para a função XMLDataSchema()

WSCERR041 / [NOTNIL_MESSAGE]

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR041 / [NOTNIL_MESSAGE]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, no momento que o client está desmontando o pacote SOAP retornado pelo serviço.

Durante a desmontagem do pacote de retorno de um Web Service, caso algum parâmetro obrigatório do serviço não esteja presente no pacote de retorno, o processamento é abortado com a mensagem acima, identificando em *[NOTNIL_MESSAGE]* o parâmetro / propriedade que não veio preenchida.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR042 / URL LOCATION não especificada.

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR042 / URL LOCATION não especificada.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, antes do envio do pacote SOAP com o(s) parâmetro(s) da ação / método solicitado.

No momento de postar o pacote SOAP de parâmetros para um Web Service, é verificada a propriedade reservada `_URL` do objeto do Serviço, que contém a URL para postagem do pacote ao servidor. Caso a mesma esteja vazia, o processamento é abortado com a mensagem acima, antes da postagem dos dados.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará `.F.` (falso), e a descrição da ocorrência deve ser recuperada através da função `GetWSError()`

Verifique o código-fonte, e certifique-se que, caso a propriedade `_URL` esteja sendo redefinida, a mesma não esteja vazia. Esta propriedade já é alimentada automaticamente pelo *engine* client de webservices, de acordo com as informações para postagem obtidas no WSDL utilizado para a geração do fonte client.

WSCERR043 / [SOAP_STYLE] SOAPSTYLE Desconhecido.

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR043 / [SOAP_STYLE] SOAPSTYLE Desconhecido.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, antes do envio do pacote SOAP com o(s) parâmetro(s) da ação / método solicitado.

No momento de postar o pacote SOAP de parâmetros para um Web Service, é verificado o formato do pacote SOAP a ser enviado ao client. Esta propriedade é definida em fonte, no momento da geração do fonte-client, e não deve ser alterada. Caso a mesma seja alterada manualmente, e não esteja num formato válido, o processamento é abortado com a mensagem acima, antes da postagem dos dados, indicando em [SOAP_STYLE] o soap style inválido informado..

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

Verifique o código-fonte, e certifique-se que o mesmo não foi alterado automaticamente pelo engine client de webservices, de acordo com as informações para postagem obtidas no WSDL utilizado para a geração do fonte client.

WSCERR044 / Não foi possível POST : URL [URP_POST]

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR044 / Não foi possível POST : URL [URP_POST]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao enviar o pacote SOAP com o(s) parâmetro(s) da ação / método solicitado.

Após montado o pacote de envio para a solicitação de processamento do serviço, o pacote é postado no servidor indicado na URL especificada no serviço. Caso o servidor de destino do pacote não seja localizado no DNS, ou não esteja no ar, o processamento é abortado com a mensagem acima, e a url de destino é especificada em [URL_POST]

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR045 / Retorno VAZIO de POST : URL <URL> ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR045 / Retorno VAZIO de POST : URL <URL> [HEADER_RET]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao enviar o pacote SOAP com o(s) parâmetro(s) da ação / método solicitado.

Apos montado o pacote de envio para a solicitação de processamento do serviço, o pacote é enviado a url discriminada no serviço. ´

Diferentemente da ocorrência WSCERR014, esta ocorrência pode ser reproduzida quando o servidor de WebServices que atendeu à requisição foi localizado, a requisição foi feita com sucesso, porém o servidor Protheus recebeu como retorno um pacote HTTP incompleto ou inválido, ou ocorreu um erro interno no servidor, referenciado no header do pacote HTTP; nestes casos o processamento é abortado com a ocorrência acima, informando em <URL> o endereço do servidor onde o dado foi postado, e, se disponível, em HEADER_RET é informado o conteúdo do Header de Retorno do HTTP.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR046 / XML Warning

[XML_WARNING] (POST em ..

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR046 / XML Warning [XML_WARNING] (POST em <URL>)

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

Apos montado e enviado o pacote de envio para a solicitação de processamento do serviço, o pacote SOAP retornado pelo serviço é analisado para a alimentação dos parâmetros Advpl . Caso seja detectada alguma inconsistência, considerada pelo parser interno de xml do Protheus como uma advertência (warning), no documento XML, o pacote SOAP de retorno é considerado inválido, e o processamento é abortado com esta ocorrência, informando em XML_WARNING a mensagem de advertência do parser interno; e em <URL> o servidor de WebServices que retornou o pacote.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR047 / XML Error

[XML_ERROR] (POST em ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR047 / XML Error [XML_ERROR] (POST em <URL>)

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

Apos montado e enviado o pacote de envio para a solicitação de processamento do serviço, o pacote SOAP retornado pelo serviço é analisado para a alimentação dos parâmetros Advpl . Caso seja detectada alguma inconsistência, considerada pelo parser interno de xml do Protheus, como um erro de sintaxe no XML, o pacote SOAP de retorno é considerado inválido, e o processamento é abortado com esta ocorrência, informando em XML_ERROR a mensagem de erro do parser interno; e em <URL> o servidor de WebServices que retornou o pacote.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError(). Veja maiores detalhes na função GetWSError(), pois ela oferece a possibilidade de recuperar os elementos principais de retorno de um pacote SOAP_FAULT isoladamente.

WSCERR048 / SOAP FAULT [FAULT_CODE] (POST em ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

***WSCERR048 / SOAP FAULT [FAULT_CODE] (POST em <URL>) :
[FAULT_STRING]***

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

Ao analisar o pacote SOAP retornado pelo serviço, para a alimentação dos parâmetros Advpl, caso o pacote de retorno contenha uma excessão do tipo *SOAP FAULT*, isto indica que houve uma falha de processamento do serviço no servidor.

O processamento é abortado com esta ocorrência, informando em [FAULT_CODE] o código da excessão SOAP, em <URL> o servidor de WebServices que retornou o pacote, e em FAULT_STRING maiores detalhes sobre a ocorrência.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR049 / SOAP RESPONSE (RPC) NOT FOUND.

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR049 / SOAP RESPONSE (RPC) NOT FOUND.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

Ao analisar o pacote SOAP retornado pelo serviço, para a alimentação dos parâmetros Advpl, caso o serviço utilize um *soapStyle = RPC*, e o node de resposta não seja encontrado no pacote, o pacote de resposta é considerado inválido, e o processamento é abortado com a mensagem acima.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR050 / SOAP RESPONSE REF <NODE_REF> (RPC) ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR050 / SOAP RESPONSE REF <NODE_REF> (RPC) NOT FOUND.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

Ao analisar o pacote SOAP retornado pelo serviço, para a alimentação dos parâmetros Advpl, caso o serviço utilize um soapStyle = RPC, e o node de resposta aponte para um outro node via referência, e este novo node não seja encontrado no pacote, o pacote é considerado inválido e o processamento é abortado com a mensagem acima, mostrando o identificador de referência não encontrado em <NODE_REF>

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR051 / SOAP RESPONSE RETURN (RPC) NOT FOUND.

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR051 / SOAP RESPONSE RETURN (RPC) NOT FOUND.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

Ao analisar o pacote SOAP retornado pelo serviço, para a alimentação dos parâmetros Advpl, caso o serviço utilize um *soapStyle = RPC*, e o node de retorno não aponte para nenhuma referência, o retorno deve estar dentro do XML, no nível do node de resposta . Caso o node de retorno não seja encontrado neste nível, o pacote de retorno é considerado inválido, e o processamento é abortado com a mensagem acima .

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR052 / Enumeration FAILED on [STRUCT_TYPE]

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR052 / Enumeration FAILED on [STRUCT_TYPE]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, antes do envio do pacote SOAP com o(s) parâmetro(s) da ação / método solicitado.

Antes da montagem do pacote SOAP, os parâmetros do método / acção solicitada do serviço são analisados e validados. Caso um parâmetro contiver uma definição de “*enumeration*”, obtida no WSDL, e for alimentado pelo fonte ‘client’ com um valor que não conste na lista de parâmetros válidos, o processamento é abortado com a mensagem acima, identificando o parâmetro envolvido em *[STRUCT_TYPE]*

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

Verifique o código-fonte client gerado em advpl, para obter a lista de parâmetros válido; e certifique-se que o parâmetro especificado está alimentado de forma correta.

WSCERR053 / WSRPCGetNode (Object) not found.

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR053 / WSRPCGetNode (Object) not found.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

Ao analisar o pacote SOAP retornado pelo serviço, para a alimentação dos parâmetros Advpl, caso o serviço utilize um *soapStyle = RPC*, no momento de análise de um retorno de uma estrutura complexa, caso o node correspondente a estrutura não seja localizado no pacote de retorno, o mesmo é considerado inválido, e o processamento é abortado com a mensagem acima.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR054 / Binding SOAP não localizado no WSDL.

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR054 / Binding SOAP não localizado no WSDL.

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE.

Durante a geração do fonte, uma vez identificado o serviço, o gerador de código procura a declaração das amarrações do serviço (*BINDINGS*) no WSDL. Dentre as amarrações encontradas, apenas são processadas aquelas que especificam o transporte de dados para o serviço no formato *SOAP*.

Caso não exista nenhuma amarração no serviço, que especifique a utilização do SOAP, o processo de geração do fonte 'client' é abortado, retornando esta ocorrência. A infraestrutura Client de WebServices do Protheus não suporta a geração de fontes-client de serviços que não utilizem pacotes *XML - SOAP* para a troca de informações.

WSCERR055 / Invalid Property Type (X) for [PARAM]

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR055 / Invalid Property Type (X) for [PARAM] (Y)

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, antes do envio do pacote SOAP com o(s) parâmetro(s) da ação / método solicitado.

Antes da montagem do pacote SOAP, os parâmetros do método / ação solicitada do serviço são analisados e validados. As propriedades da classe, utilizadas como parâmetros, devem ser alimentadas com os tipos Advpl apropriados, de acordo com sua definição. Caso uma determinada propriedade *[PARAM]* do objeto 'Client' do serviço esteja alimentada com um tipo de dado Advpl *[X]*, porém o tipo esperado era *[Y]*, o processamento é abortado com a ocorrência de erro acima.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

Verifique o código-fonte client gerado em advpl, e certifique-se que o parâmetro especificado está sendo alimentado de forma correta, com o tipo apropriado.

WSCERR056 / Invalid XML-Soap Server Response : ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR056 / Invalid XML-Soap Server Response : soap-envelope not found.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao iniciar o processamento do pacote SOAP recebido como retorno da ação / método solicitado.

Ao analisar o pacote SOAP retornado pelo serviço, caso o mesmo não contenha um envelope (*soap-Envelope*) de resposta, o retorno é considerado inválido, e o processamento é abortado com a mensagem acima .

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR057 / Invalid XML-Soap Server Response : ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR057 / Invalid XML-Soap Server Response : soap-envelope empty.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao iniciar o processamento do pacote SOAP recebido como retorno da ação / método solicitado.

Ao analisar o pacote SOAP retornado pelo serviço, caso não seja possível determinar o prefixo do SOAP Envelope utilizado, o retorno é considerado inválido, e o processamento é abortado com a mensagem acima .

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR058 / Invalid XML-Soap Server Response : ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR058 / Invalid XML-Soap Server Response : Invalid soap-envelope [SOAP_ENV] object as valtype [X]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao iniciar o processamento do pacote SOAP recebido como retorno da ação / método solicitado.

Ao analisar o pacote SOAP retornado pelo serviço, caso o *soap-envelope* determinado *[SOAP_ENV]*, esperado como um Objeto, foi recebido com um tipo *Advpl [X]*. Isto invalida o pacote soap recebido, sendo o processamento abortado com a ocorrência acima.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR059 / Invalid XML-Soap Server Response : ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR059 / Invalid XML-Soap Server Response : soap-body not found.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao iniciar o processamento do pacote SOAP recebido como retorno da ação / método solicitado.

Semelhante a ocorrência WSCERR056, esta ocorrência indica que não foi possível determinar o corpo (*soap-body*) do pacote SOAP retornado pelo serviço; o que invalida o pacote de retorno, sendo o processamento abortado com esta ocorrência de erro.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR060 / Invalid XML-Soap Server Response : ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR060 / Invalid XML-Soap Server Response : soap-body envelope empty.

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao iniciar o processamento do pacote SOAP recebido como retorno da ação / método solicitado.

Semelhante a ocorrência WSCERR057, esta ocorrência indica que pacote SOAP retornado, não foi possível determinar o prefixo do corop (*soap-body*) utilizado; o que invalida o pacote de retorno, sendo o processamento abortado com esta ocorrência de erro.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR061 / Invalid XML-Soap Server Response : ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR061 / Invalid XML-Soap Server Response : Invalid soap-body [BODY] object as valtype [TYPE]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao iniciar o processamento do pacote SOAP recebido como retorno da ação / método solicitado.

Semelhante a ocorrência WSCERR058, esta ocorrência indica que no *SOAP* retornado, o corpo (*soap-body*) determinado [*BODY*], esperado como um Objeto, foi recebido como um tipo Advpl [*TYPE*], ; o que invalida o pacote de retorno, sendo o processamento abortado com esta ocorrência de erro.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR062 / Invalid XML-Soap Server Response : ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR062 / Invalid XML-Soap Server Response : Unable to determine Soap Prefix of Envelope [SOAP_ENV]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao iniciar o processamento do pacote *SOAP* recebido como retorno da ação / método solicitado.

Esta ocorrência indica que, no *SOAP* retornado, o envelope (*soap-envelope*) determinado [*SOAP_ENV*], não está em um formato que seja possível determinar o nome do envelope; o que invalida o pacote de retorno, sendo o processamento abortado com esta ocorrência de erro.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR063 / Argument error : Missing field [NODE]

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR063 / Argument error : Missing field [NODE] as [TYPE]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao iniciar a montagem do pacote SOAP com os parâmetros para a chamada do serviço.

Esta ocorrência indica que, o parâmetro obrigatório determinado em [NODE], com o tipo [TYPE], não foi alimentado para a chamada da função 'client'. Esta ocorrência invalida a montagem do pacote de envio, abortando o processamento antes do envio do pacote, com esta ocorrência.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR064 / Invalid Content-Type return (HTTP_HEAD)

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR064 / Invalid Content-Type return (HTTP_HEAD) from <URL>

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado. Após montado e enviado o pacote de envio para a solicitação de processamento do serviço, o pacote SOAP retornado pelo serviço é analisado para a alimentação dos parâmetros Advpl .

Esta ocorrência indica que, o header HTTP de retorno do serviço, postado em <URL>, veio com o conteúdo do header HTTP retornado pelo servidor, indica o uso de *content-type* diferente de *XML*, o que invalida o processamento do retorno. Um Web Service 'client' sempre espera por um pacote de retorno com um '*Content-type: text/xml*' de um Web Services SERVER.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

Esta ocorrência normalmente é reproduzida, quando um determinado WebService não está mais publicado no endereço especificado, porém a url ainda é válida. De modo que, ao receber a requisição, o servidor devolve uma página *HTML*, com uma mensagem do tipo '*Page not Found*'.

WSCERR065 / EMPTY Content-Type return (HEADER) ...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR065 / EMPTY Content-Type return (HEADER) from <URL>

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

Semelhante a ocorrência WSCERR064, esta ocorrência indica que, após a postagem de um pacote SOAP ao servidor de destino do WebService, em <URL>, o conteúdo do header Http retornado (HEADER) retornado pelo servidor, não possuía a identificação do *Content-Type*, o que invalida o processamento de retorno. O client Advpl sempre espera por um pacote de resposta com um *content-type: text/xml* como retorno.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

WSCERR066 / Invalid INVALID WSDL Content-Type (...)

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR066 / Invalid INVALID WSDL Content-Type (HTTP_HEAD) from <URL>

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE.

Esta ocorrência indica que, o header HTTP de retorno da requisição do WSDL, solicitado no endereço <URL>, veio identificando um tipo de documento (*content-type*) diferente de text/plain ou text/xml, o que invalida o processamento do retorno. Um Web Service 'client' sempre espera por um pacote de retorno com um 'Content-type: text/xml' ou 'text/plain', de um Web Services SERVER.

Esta ocorrência normalmente é reproduzida, quando um determinado WebService não está mais publicado no endereço especificado, porém o serviço de http ainda está ativo no servidor solicitado. De modo que, ao receber a requisição, o servidor devolve uma página HTML, com uma mensagem do tipo 'Page not Found'.

WSCERR067 / EMPTY WSDL Content-Type (HTTP_HEAD)

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR067 / EMPTY WSDL Content-Type (HTTP_HEAD) from <URL>

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE.

Esta ocorrência indica que, o header HTTP de retorno do WSDL, solicitado através do link <URL>, veio com o conteúdo do header HTTP sem a informação do tipo de conteúdo do documento (*content-type*). Um documento WSDL deve ser retornado pelo servidor de WebServices, informando no header HTTP um tipo de documento (*content-type*) definido como *text/plain* ou *text/xml*

WSCERR068 / NOT XML SOURCE from <URL>

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR068 / NOT XML SOURCE from <URL>

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE.

Esta ocorrência indica que, o documento retornado pelo servidor de webservices não se trata de um XML válido para ser analisado. O documento WSDL deve sempre iniciar com o node da declaração do XML (*<?XML ...*) . Caso não possua esta informação, o primeiro node deve obrigatoriamente ser a definição do serviço (*<DEFINITIONS*). Se o documento WSDL retornado não atender à estes requisitos, o processamento é abortado com a mensagem acima.

WSCERR069 / BYREF [PARAM] WITH NO INPUT ARGUMENT :

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR069 / BYREF [PARAM] WITH NO INPUT ARGUMENT : UNSUPPORTED WEBSERVICE

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE.

Quando da geração do fonte, caso o WSDL retornado informe um método de Web Services, que possua mais de um parâmetro de retorno, isto caracteriza um método que trabalha com parâmetros por referência (*BYREF*). Neste caso, após o cruzamento dos retornos do método com os parâmetros, deve restar no máximo um retorno. Caso mesmo assim, reste mais de um retorno, o WSDL é considerado inválido, sendo o processo de geração abortado com a mensagem de erro acima, informando em [PARAM] o retorno excedente, que deveria ser localizado nos parâmetros.

WSCERR070 / Requisição HTTPS não suportada neste..

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR070 / Requisição HTTPS não suportada neste BUILD [PROTHEUS_BUILD]

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, antes do envio do pacote SOAP com o(s) parâmetro(s) da ação / método solicitado.

No momento de postar o pacote SOAP de parâmetros para um Web Service, é verificado se o protocolo em uso é o HTTPS; e se o mesmo já é suportado pelo Build atual do servidor Protheus em uso.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSError()

Verifique o código-fonte, e certifique-se que, caso a propriedade _URL esteja sendo redefinida, a mesma não esteja sendo redefinida para um endereço utilizando HTTPS. Caso a propriedade _URL não esteja sendo re-definida, e o serviço solicitado exija o envio dos dados através de HTTPS, o build do servidor Protheus deve ser atualizado.

WSCERR071 / INVALID HTTP HEADER (HTTPHEAD) from...

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR071 / INVALID HTTP HEADER (HTTPHEAD) from <URL>

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE.

Quando da geração de Códigos fonte Advpl, caso o servidor informado, acessado via URL, retorne um pacote HTTP, com um header de retorno que não seja identificado como HTTP, o processo de geração é abortado com a ocorrência acima, informando em *<httphead>* o header informado, e em *<url>* o endereço informado para a solicitação do WSDL.

Dentre as possíveis causas, podemos considerar que a URL informada não corresponde a um servidor HTTP ou de WEB SERVICES. Para certificar-se da ocorrência, abra a URL especificada utilizando um Web Browser.

WSCERR072 / HTTP REQUEST ERROR (HEADER) from <URL>

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR072 / HTTP REQUEST ERROR (HEADER) from <URL>

Esta ocorrência de erro é reproduzida, quando da geração de um fonte de WebServices 'Client', utilizando o Protheus IDE.

Quando da geração de Códigos fonte Advpl, caso o servidor informado, acessado via URL, retorne um pacote HTTP, com um header de retorno HTTP, porém com um status diferente de 200 (OK) , o processo de geração é abortado com a ocorrência acima, informando em <HEADER> a primeira linha do cabeçalho HTTP retornado, e em <url> o endereço informado para a solicitação do WSDL.

Dentre as prováveis causas, podemos considerar os status de retorno '403 Forbidden', retornados por Proxys que requerem autenticação ou não permitem o acesso à url especificada, o '500 Internal Server Error', que indica uma ocorrência interna de erro no servidor, que impossibilitou o retorno do WSDL.

WSCERR073 / Build (BUILD) XML Internal Error

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERR073 / Build (BUILD) XML Internal Error

Esta ocorrência é reproduzida, quando da utilização de um fonte Client de WebServices, ao processar o pacote SOAP recebido como retorno da ação / método solicitado.

O pacote SOAP retornado pelo serviço é analisado para a alimentação dos parâmetros Advpl. em primeiro momento, são realizadas as consistências de cabeçalho de protocolo (header) , e em seguida o pacote SOAP é desmontado por um parser interno do Protheus, onde é verificada a sintaxe do documento XML (Veja ocorrências WSCERR046 e WSCERR047), e a resultante deste processo será um objeto intermediário.

Se e somente se, o conteúdo SOAP retornado pelo serviço, contenha um erro estrutural ou sintático, que não seja detectado pelo parser interno como um erro ou advertência, este objeto intermediário não é gerado, o que impossibilita a rotina de prosseguir o processamento. Esta ocorrência já foi reproduzida anteriormente, em builds do Protheus anteriores à Dezembro/2003. Em releases posteriores a este, o tratamento dos pacotes de retorno do serviço foi revisado; desde então esta ocorrência não mais foi reproduzida.

Esta ocorrência é capturada pelo próprio fonte do método, sendo que o método 'Client' chamado retornará .F. (falso), e a descrição da ocorrência deve ser recuperada através da função GetWSCError()

WSCERRINT / [ERROR_DESCRIPTION]

Revisão: 29/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSCERRINT / [ERROR_DESCRIPTION]

Quando executado um método 'Client' de WebServices, as ocorrências de falha dentro destas worinas são protegidas por um tratamento de erro exclusivo, que informa detalhes da ocorrência.

Se, e somente se, o tratamento de erro for acionado por uma ocorrência inesperada, em algum ponto do processamento do método da classe Client, a descrição da ocorrência de erro é capturada, e mostrada em <ERROR_DESCRIPTION> , e a ocorrência é prefixada com o código WSCERRINT (*Web Services Client Internal Error*)

Caso seja reproduzida esta ocorrência, verifique os parâmetros informados ao método chamado, e certifique-se que o código fonte da classe 'Client' em Advpl não sofreu nenhuma alteração manual, após a geração do próprio.

Terminologias / Glossário

Revisão: 28/04/2004

Abrangência

Versão 7.10

Versão 8.11

WSDL : (*Web Services Description Language*) : Trata-se de um documento, em formato de acordo com as definições de Web Services, através do qual um provedor de um serviço provê a discriminação detalhada das funcionalidades de um serviço. Este documento em geral é fornecido através de uma URL, apontando para o servidor que provê o serviço. Utilizando este documento, o Protheus é capaz de gerar automaticamente um 'Fonte Client' para estabelecer a conexão e utilização do serviço, através da geração de uma classe 'Client' em Advpl.

Web Service 'Client' : Aplicação desenvolvida à partir de uma definição (WSDL) publicada e disponibilizada por uma aplicação 'Server'.

Web Service 'Server' : Aplicação desenvolvida para tornar disponível um recurso, processamento ou informação, juntamente com sua definição (WSDL), para tornar possível o desenvolvimento de uma aplicação 'Cliente' que irá solicitar a execução da aplicação em si.

'Fonte Client' : Código fonte Advpl, gerado pela ferramenta do IDE 'Gerar Cliente WebServices...', a partir de uma definição WSDL publicada em um servidor HTTP ou disponibilizada em um arquivo .WSDL.

'Tipo básico' : São chamados de tipos básicos, uma lista de tipos de informações 'nativa' ,implementada na definição dos WebServices.

'Estrutura' : É chamada de estrutura, uma classe intermediária de dados para Web Services, cuja função é definir uma informação que consiste no agrupamento de outras informações e/ou estruturas.