

Terminologias / Glossário

Revisão: 16/06/2005

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'. A aplicação 'client' deve ser capaz de se comunicar com a aplicação Server, utilizando mensagens em XML/SOAP, segundo as regras fornecidas pelo Web Services Server através do WSDL.

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 'Client' que irá solicitar a execução da aplicação server e obter o retorno desta.

'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.

'SOAP' - Sigla de Simple Object Access Protocol, ou protocolo simples de acesso a objetos. O SOAP é um padrão aberto, baseado em XML, criado pela Microsoft, Ariba e IBM para padronizar a transferência de dados entre aplicações. Pode ser usado em combinação com vários outros protocolos comuns da Internet, como HTTP e SMTP.

'XML' - Sigla de Extensible Markup Language, o XML é uma linguagem baseada em tags semelhante ao HTML. Sua principal característica é a extensibilidade. Quem emite um documento XML pode criar tags personalizadas, que são definidas num documento anexo, que tem extensão XSD.

'XSD' - Sigla de XML Schema Definition. Arquivo associado a um documento XML que descreve e valida os dados no documento, permitindo a criação de tipos de dados personalizados e regras específicas para os mesmos.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / **04. Tipos Básicos de Dados - 'Server'**](#)

[Topo da Página](#)

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.

Grupos Relacionados

| | |
|---|---|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos |
|---|---|

Veja também

| | |
|---|---|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSCLIENT |
|---|---|

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.

Grupos Relacionados

| | |
|---|---|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos |
|---|---|

Veja também

| | |
|---|--|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSSERVICE |
|---|--|

[Topo da Página](#)

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.

Grupos Relacionados

| | |
|---|---|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos |
|---|---|

Veja também

| | |
|---|--|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSSTRUCT |
|  | Principal / A Tecnologia Protheus / Web Services / Aplicações Server / 05. Estruturas - Tipos complexos |

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Comandos](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Comandos / **ENDWSCLIENT**](#)



[Principal / A Tecnologia Protheus / Web Services / Comandos / **WSDATA**](#)



[Principal / A Tecnologia Protheus / Web Services / Comandos / **WSSERVICE**](#)

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

WSDATA cVarName AS [ARRAY OF] cVarType [OPTIONAL]

Parâmetros

| Argumento | Tipo | Descrição |
|-----------|-----------|---|
| cVarName | Character | cVarName corresponde ao nome da propriedade a declarar. |
| AS | Character | Separador para indicar o tipo da propriedade. |
| ARRAY OF | Character | 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 | Character | 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 | Character | 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.

Grupos Relacionados[Principal / A Tecnologia Protheus / Web Services / Comandos](#)**Veja também**

| | |
|---|---|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSCLIENT |
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSSERVICE |
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSSTRUCT |
|  | Principal / A Tecnologia Protheus / Web Services / Aplicações Server / 04. Tipos Básicos de Dados - 'Server' |

WSMETHOD

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Comandos](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / **06. Métodos 'Server' em Advpl - Características**](#)

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) desta classe ...)

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

Grupos Relacionados

| | |
|---|---|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos |
|---|---|

Veja também

| | |
|---|---|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / ENDWSSERVICE |
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSCLIENT |
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSDATA |

WSSTRUCT

Revisão: 22/04/2004

Abrangência

| | |
|-------------|-------------|
| Versão 7.10 | Versão 8.11 |
|-------------|-------------|

Sintaxe

WSSTRUCT cStructName

Parâmetros


| Argumento | Tipo | Descrição |
|-------------|----------|---|
| cStructName | Caracter | 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.

Grupos Relacionados

| | |
|---|---|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos |
|---|---|

Veja também

| | |
|---|--|
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / ENDWSSTRUCT |
|  | Principal / A Tecnologia Protheus / Web Services / Comandos / WSDATA |
|  | Principal / A Tecnologia Protheus / Web Services / Aplicações Server / 05. Estruturas - Tipos complexos |

Web Services - Visão Geral

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Introdução

Visto de forma simples, um Web Service é uma classe publicada em um servidor de aplicação, para acesso externo. De modo que, um programa de software "client" irá solicitar ao "server" um processamento de uma ação / método do serviço, enviando o(s) parâmetro(s) ao mesmo através de uma string XML, formatada no padrão SOAP (Simple Object Access Protocol), e o serviço retornará ao client uma string XML de resposta, também formatada no padrão SOAP.

Diferentemente dos Web Sites, que são apresentações de dados feitas para serem vistas em um browser por um internauta, um Web Service é construído para ser acessado diretamente por outro serviço ou programa de software. Os serviços podem ser combinados entre si e com outras aplicações para construir serviços interativos e inteligentes.

A universalidade de um Web Service, entre outros fatores, reside justamente na maneira de disponibilizar uma espécie de "manual" do serviço, chamado de WSDL (Web Services Description Language), que nada mais é do que uma descrição detalhada do serviço, que é capaz de resumir em um único documento os métodos de um serviço, os parâmetros, retornos e detalhes do mesmo. Um documento WSDL é de tal forma completo que praticamente elimina a necessidade de qualquer tipo de contato direto entre o desenvolvedor de um Serviço (server) e o analista que irá montar um client para interagir com ele. Também foi idealizado o uso do protocolo HTTP originalmente para a troca de pacotes de requisição e retorno de processamento, pois a maioria dos sistemas operacionais têm recursos para suportá-lo ; e foi escolhido o formato SOAP por ser um padrão aberto, criado pela Microsoft, Ariba e IBM para padronizar a transferência de dados em diversas aplicações, que utiliza como veículo o universalmente conhecido XML (Extended Markup Language) .

Principais usos

As aplicações utilizando Web Services têm um leque de possibilidades de uso muito grande. Atualmente utiliza-se Web Services para a troca de dados / integração entre sistemas multi-plataforma ou não, disponibilização de ferramentas e dados para consulta remota, integrações B2B e B2C, enfim : Todo e qualquer processamento que seja interessante disponibilizar de uma forma mais abrangente.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services](#)

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
```

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Exemplos Advp!](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Funções / **GETWSCVER**](#)

[Topo da Página](#)

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
```

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Exemplos Advpl](#)

[Topo da Página](#)

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 à 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  
(...)
```

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Exemplos Advpl](#)

Veja também



FUNÇÕES

GETWSCERROR

Revisão: 22/04/2004

Abrangência

Versão 7.10

Versão 8.11

Sintaxe

GETWSCERROR ([nInfo]) --> xErrorInfo

Parâmetros

| Argumento | Tipo | Descrição |
|-----------|----------|---|
| nInfo | Numérico | nInfo especifica qual informação pertinente ao erro deve ser retornada, podendo ser : 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 GetWSCError(), 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 .

Exemplos

 [Exemplo de uso da função GETWSCERROR](#)

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Funções](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente / Ocorrências de Erro / **WSCERR048 / SOAP FAULT \[FAULT_CODE\] \(POST em ...**](#)

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'.

Exemplos

 [Exemplo de uso da função GETWSCVER](#)

Grupos Relacionados

| | |
|---|--|
|  | Principal / A Tecnologia Protheus / Web Services / 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'.

Grupos Relacionados

| | |
|---|--|
|  | Principal / A Tecnologia Protheus / Web Services / 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 HHeader 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. |
| 6 | SOAPFAULT_RECEIVER | 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'


Exemplos

 [Exemplo de uso da função SETSOAPFAULT](#)

Grupos Relacionados

| | |
|---|--|
|  | Principal / A Tecnologia Protheus / Web Services / Funções |
|---|--|

Veja também

| | |
|---|--|
|  | Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente / Ocorrências de Erro / WSCERR048 / SOAP FAULT [FAULT_CODE] (POST em ... |
|---|--|

[Topo da Página](#)

SOAPDTGETD

Revisão: 22/04/2004

Abrangência

Versão 8.11

Sintaxe

SOAPDTGETD (< cDateTime >) --> dDate

Parâmetros

| Argumento | Tipo | Descrição |
|-----------|----------|--|
| cDateTime | Caracter | 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*.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Funções](#)

SOAPDTGETT

Revisão: 22/04/2004

Abrangência

Versão 8.11

Sintaxe

SOAPDTGETT (< cDateTmie >) --> cTime

Parâmetros

| Argumento | Tipo | Descrição |
|-----------|----------|--|
| cDateTmie | Caracter | String, no formato "Soap" DateTime, a ser considerada. |

Retorno

| Tipo | Descrição |
|----------|---|
| Caracter | 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

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / 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*.

Grupos Relacionados

| | |
|---|--|
|  | Principal / A Tecnologia Protheus / Web Services / Funções |
|---|--|

[Topo da Página](#)

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 AdvPI, 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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Funções](#)

[Topo da Página](#)

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'.

Grupos Relacionados



[Principal](#) / [A Tecnologia Protheus](#) / [Web Services](#) / [Funções](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

[Topo da Página](#)

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 "Querys" utilizadas no método, filtros setados em tabelas principais, eu 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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Comandos / **WSDATA**](#)



[Principal](#) / [A Tecnologia Protheus](#) / [Web Services](#) / **[Web Services com Protheus](#)**

[Topo da Página](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Comandos / **ENDWSSTRUCT**](#)



[Principal / A Tecnologia Protheus / Web Services / Comandos / **WSSTRUCT**](#)

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 à '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.**

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

Veja também



[Principal](#) / [A Tecnologia Protheus](#) / [Web Services](#) / [Comandos](#) / **[WSMETHOD](#)**

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".

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

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 SetSoapFaut().

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente / Ocorrências de Erro / **WSCERR048 / SOAP FAULT \[FAULT_CODE\] \(POST em ...**](#)

09. Serviço de Exemplo : SERVERTIME

Revisão: 16/05/2008

Abrangência

| | | |
|-------------|-------------|-------------|
| Versão 10.1 | 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 Horário          as String
6          WSMETHOD GetServerTime
7      ENDWSSERVICE
8
9      WSMETHOD GetServerTime WSRECEIVE NULLPARAM WSEND Horário
WSSERVICE SERVERTIME
10     ::Horário := 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 <i>Web Services</i> . Ele é obrigatório para o desenvolvimento de WebServices. |
| Linha 4 | Com esta instrução, é definido o início 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 <u>Horario</u> (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 constataremos 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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente / **Aplicações Protheus 'Client' de WebServices**](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

[Topo da Página](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

[Topo da Página](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

[Topo da Página](#)

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

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

Grupos Relacionados



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

Grupos Relacionados



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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

[Topo da Página](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Falhas de Carga dos Serviços](#)

[Topo da Página](#)

OCORRÊNCIAS DE ERRO FATAL

Ocorrências de erro fatal e tratamento de erro

Revisão: 26/04/2004

Abrangência

Versão 7.10

Versão 8.11

As ocorrências de erro fatal relacionadas nesta seção referem-se à falhas de carga do engine de WebServices como um todo e ocorrências de falha de processamento quando da chamada para execução dos serviços.

Em cada documento, é discriminado um resumo da ocorrência, sua possível causa e possíveis soluções.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

[Topo da Página](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

[Topo da Página](#)

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*.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

[Topo da Página](#)

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'

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

[Topo da Página](#)

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 à '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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

REQUIRED Return property [X] Type [Y] Unexpect ...

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

[Topo da Página](#)

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,

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

Return property [X] AS OBJECT Type [Y] Unexpect ..

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

[Topo da Página](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

[Topo da Página](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / 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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / Ocorrências de Erro Fatal](#)

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente](#)

Veja também



[Principal / A Tecnologia Protheus / Web Services / Aplicações Server / 09.](#)
Serviço de Exemplo : SERVETIME

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, apos 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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente](#)

Geração de Client em Advpl - Passo 02

Revisão: 30/04/2004

Abrangência

Versão 7.10

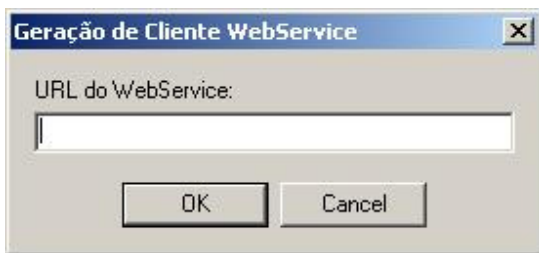
Versão 8.11

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.

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente](#)

[Topo da Página](#)

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 *Web Service*
 Linha 4 Uma variável local é declarada para conter o Objeto do *Web Service* ‘*client*’
 Linha 6 Utilizando-se do serviço, a variável oSvc é alimentada com uma nova instância do *Web Services* ‘*client*’, 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 ‘*client*’ 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 armazenado 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

Grupos Relacionados



Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente

[Topo da Página](#)

Geração de Client em Advpl - Passo 04

Revisão: 22/04/2004

Abrangência

Versão 7.10

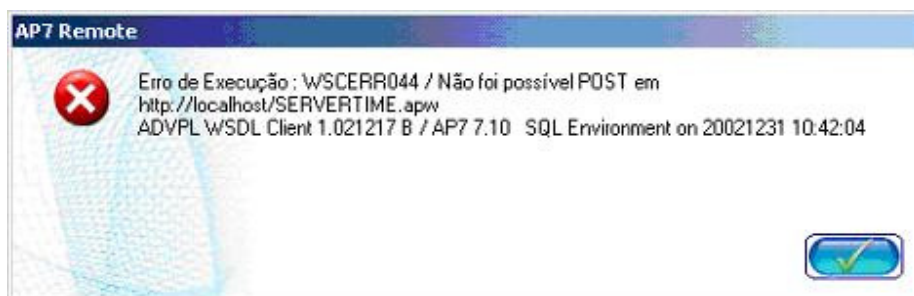
Versão 8.11

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 :



Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente](#)

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:soap='http://schemas.xmlsoap.org/soap/envelope/'> <soap:Body> <GETSERVERTIME  
RESPONSE
```

```
xmlns='http://automan:8000/webservice/'> <GETSERVERTIMERESULT> 10:37:10<  
</GETSERVERTIMERESULT> </GETSERVERTIMERESPONSE> </soap:Body> </soap:Envelope>
```

Fim Thread (siga0984, 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 peço Web Services 'client' Advpl estão relacionadas no Tópico Web Services 'client' – Códigos de Erro .

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente](#)

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

Grupos Relacionados



[Principal / A Tecnologia Protheus / Web Services / Aplicações Cliente](#)