
NCPSOAP: Thistle NCP Enabling HTTP
Communication and SOAP Wrapping

CML00017-01

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com
Copyright © 2014 by Code Magus Limited
All rights reserved

Contents

1	Thistle Interface	2
1.1	Introduction	2
1.2	<i>Thistle</i> Interface Definition for NCP SOAP	2
2	NCP SOAP NCP Interface reference	2
2.1	NCP SOAP NCP Interface Methods	2
2.2	open()	2
2.2.1	Parameters for Function open()	2
2.3	send()	3
2.3.1	Parameters for Function send()	3
2.4	receive()	3
2.5	NCP SOAP NCP Interface Attributes	3
2.6	EnterpriseContext	4
2.7	Identity	4
2.8	AuthenticationLOC	4
2.9	ExpiryDateTime	4
2.10	IdentifierType	4
2.11	TicketSignature	4
2.12	SOAPAction	4
2.13	Method	5
2.14	URI	5
2.15	ProxyURI	5
2.16	SOAPEnvelope	5
2.17	PersistentConnection	5
2.18	SendEncapsulated	5
2.19	SendEncapsulatedName	5
2.20	ReceiveEncapsulatedName	6
2.21	Example <i>Thistle</i> Script Using NCP SOAP	6

1 Thistle Interface

1.1 Introduction

The Code Magus Limited NCP SOAP DLL provides standard NCP functions to send an XML encoded buffer as a SOAP RPC using HTTP as the transport medium and to receive an XML encoded buffer in reply.

This document shows how the NCP SOAP functions can be used by the *Thistle* scripts.

1.2 *Thistle* Interface Definition for NCP SOAP

The *Thistle* Interface Definition for NCP SOAP is a standard NCP Interface to *Thistle*.

2 NCP SOAP NCP Interface reference

2.1 NCP SOAP NCP Interface Methods

The following methods are exposed by the NCP SOAP NCP Interface. These are shown here together with the names of their parameters.

2.2 `open ()`

- `open ()`

Function `open ()` is the NCP function to open a circuit.

This function creates a circuit and an HTTP session using parameters passed to the function.

2.2.1 Parameters for Function `open ()`

- `URI`: The `URI` parameter contains the URN encoded name of the SOAP endpoint. It is a mandatory parameter.
- `SOAPEnvelope`: The `SOAPEnvelope` parameter has valid values of `yes` or `no` and determines whether the XML document will be wrapped in a SOAP envelope. This parameter is optional and defaults to `yes`.
- `SOAPAction`: The `SOAPAction` parameter contains the name of the SOAP function to be executed. This parameter is optional.

- `ProxyURI`: The `ProxyURI` parameter contains the URI of the HTTP Proxy host. This parameter is optional.
- `PersistentConnection`: The `PersistentConnection` parameter has valid values of yes or no and determines whether the HTTP connection is persisted for more than one request. This parameter is optional and defaults to yes.
- `SendEncapsulated`: The `SendEncapsulated` parameter has valid values of yes or no and determines whether the first XML element in the send buffer is an encapsulating element that should be removed before the document is inserted into the SOAP envelope. This parameter is optional and defaults to no.
- `SendEncapsulatedName`: The `SendEncapsulatedName` parameter defines the name of the encapsulating element. An error will be returned if the first element does not have this name. This parameter has no effect if `SendEncapsulated` is set to no. This parameter is optional.
- `ReceiveEncapsulatedName`: The `ReceiveEncapsulatedName` parameter specifies the name of the first element in the received document, if the document must be encapsulated. This parameter is optional.

2.3 `send()`

- `send()`

Function `send()` sends a buffer which must contain a valid XML Document unless `SOAPEnvelope=NO` is specified.

2.3.1 Parameters for Function `send()`

- `buffer` The `buffer` parameter contains the request buffer to be sent which must contain a valid XML document unless `SOAPEnvelope=NO` is specified.

2.4 `receive()`

- `receive()`

Function `receive()` returns the response buffer in reply to the request.

2.5 NCP SOAP NCP Interface Attributes

The following attributes are exposed by the NCP SOAP NCP Interface..

2.6 EnterpriseContext

- **EnterpriseContext**: This attribute can have the value YES or NO indicating whether an extended SOAP Header must be added to the SOAP Envelope.

2.7 Identity

- **Identity**: This attribute is the Ticket Identity and is relevant only when EnterpriseContext=YES.

2.8 AuthenticationLOC

- **AuthenticationLOC**: This attribute is the Ticket AuthenticationLOC and is relevant only when EnterpriseContext=YES.

2.9 ExpiryDateTime

- **ExpiryDateTime**: This attribute is the Ticket ExpiryDateTime and is relevant only when EnterpriseContext=YES.

2.10 IdentifierType

- **IdentifierType**: This attribute is the Ticket IdentifierType and is relevant only when EnterpriseContext=YES.

2.11 TicketSignature

- **TicketSignature**: This attribute is the Ticket TicketSignature and is relevant only when EnterpriseContext=YES.

2.12 SOAPAction

- **SOAPAction**: This attribute is the SOAPAction HTTP Header that will be sent to the host.

2.13 Method

- **Method**: This attribute is the HTTP Method (default POST).

2.14 URI

- **URI**: This attribute is the HTTP URI which identifies the SOAP endpoint.

2.15 ProxyURI

- **ProxyURI**: This attribute is the URI which identifies the HTTP Proxy.

2.16 SOAPEnvelope

- **SOAPEnvelope**: This attribute has valid values of yes or no and determines whether the NCP will attempt to wrap the send buffer in a SOAP envelope.

2.17 PersistentConnection

- **PersistentConnection**: This attribute has valid values of yes or no and determines whether the HTTP connection is persisted for more than one request.

2.18 SendEncapsulated

- **SendEncapsulated**: The SendEncapsulated attribute has valid values of yes or no and determines whether the first XML element in the send buffer is an encapsulating element that should be removed before the document is inserted into the SOAP envelope.

2.19 SendEncapsulatedName

- **SendEncapsulatedName**: The SendEncapsulatedName attribute defines the name of the encapsulating element. An error will be returned if the first element does not have this name. This attribute has no effect if SendEncapsulated is set to no.

2.20 ReceiveEncapsulatedName

- **ReceiveEncapsulatedName**: The `ReceiveEncapsulatedName` attribute specifies the name of the first element in the received document, if the document must be encapsulated.

2.21 Example *Thistle* Script Using NCP SOAP

The following scripts shows how the NCP SOAP interface could be used for processing SOAP Requests:

```
package NCPSOAPSamplePackage;
  created by    'Sreemanth';
  description  'Validation Engine Customer Change message';
  date        2008-05-04T13:13:13;
  target      'SOAP NCP';
  interface   Portal.NCP                : CodeMagus.NCP;
  interface   Portal.Excel              : CodeMagus.Excel;
  interface   Portal.XML                 : CodeMagus.XML;

begin
  ncppath1 := "C:\\build\\build\\bin\\ncpsoap.dll";
  URI := "http://soadev2:9081/ValidationEngineServiceWS/services/ValidationEnginePort";
  SOAPAction := "";
  {http://contracts.it.nednet.co.za/services/customer-services/2007-07-21/Customer/Cus
  ProxyURI := "";
  PersistentConnection := "YES";
  SendEncapsulated := "NO";
  SendEncapsulatedName := "";
  ReceiveEncapsulatedName := "";

  xml := Portal.XML.Connect();
  ncp := Portal.NCP.Connect(ncppath1);

  ncp.open (URI:=URI,
           SOAPAction:=SOAPAction,
           ProxyURI:=ProxyURI,
           PersistentConnection:=PersistentConnection,
           SendEncapsulated:=SendEncapsulated,
           SendEncapsulatedName:=SendEncapsulatedName,
           ReceiveEncapsulatedName:=ReceiveEncapsulatedName);

  with document_1 do begin
    with validate[00] do begin
      with input[00] do begin
        document.value := "Error";
      end
    end
  end

end
```

```

System.DumpScope (ncp);

System.DumpScope (document_1);

data.request := xml.GetDocument (document_1);

ncp.send (data.request);

data.reply := System.TranslateFromASCIIToString (ncp.receive ());

data.actual := xml.GetTree (data.reply);
System.DumpScope (data);

end.

```

Running this script produces the expected output:

```

Initializing run...
cannot find thistle config file
CodeMagus.NCP
CodeMagus.Excel
CodeMagus.XML
ncp
  send<method>
  open<method>
  receive<method>
EnterpriseContext <property:string len 2> = 'no'
SOAPAction <property:string len 0> =
Identity <property:string len 0> =
AuthenticationLOC <property:string len 0> =
SessionSequenceNr <property:string len 0> =
ExpiryDateTime <property:string len 0> =
IdentifierType <property:string len 0> =
TicketSignature <property:string len 0> =
Method <property:string len 4> = 'POST'
URI <property:string len 75> =
  'http://soadev2:9081/ValidationEngineServiceWS/services/ValidationEngineP'
  'ort'
ProxyURI <property:string len 0> =
PersistentConnection <property:string len 3> = 'YES'
SendEncapsulated <property:string len 2> = 'no'
SendEncapsulatedName <property:string len 0> =
ReceiveEncapsulatedName <property:string len 0> =
document_1
  validate
    00
    input
      00
      document
        value <property:string len 5> = 'Error'
Unknown error: Line 1, char 38: The prefix 'soapenv' has not been mapped to any URI
data

```



```
actual <none>
reply <property:string len 227> =
  '<?xml version="1.0" ?><soapenv:Fault><faultcode>soapenv:Server.generalEx'
  'ception</faultcode><faultstring><![CDATA[com.codemagus.validation.Validati'
  'onEngineException: No namespace found in document]]></faultstring></soap'
  'env:Fault?>'
request <property:string len 117> =
  '<?xml version="1.0" encoding="UTF-8" standalone="no" ?><validate><input>'
  '<document>Error</document></input></validate>'
Run completed...
```