



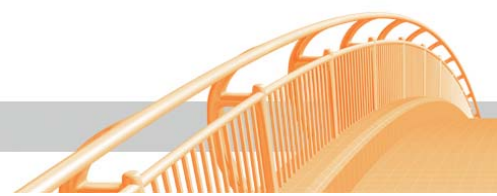
Comtarsia Web Client 2006

Introduction and Installation

Version: 1.2.5.4, 04-Jul-2006

Table of contents

1.	Comtarsia Web Sync Client.....	3
1.1	Introduction	3
1.2	Functionality	3
1.2.1	HTML Interface.....	5
1.2.2	SOAP-API	5
1.3	Installation	5
1.3.1	General	5
1.3.2	Installation under IIS 5.....	8
1.3.3	Installation under IIS 6.....	11
1.3.4	Configuring the WebClient.....	19
1.3.5	The HTML Interface.....	20
1.3.6	SOAP-API	22
1.4	Examples for SOAP Client-Development	23
1.4.1	PHP SOAP-Client.....	23
1.4.2	Java SOAP-Client	24



1. Comtarsia Web Sync Client

1.1 Introduction

The Comtarsia Web Sync Client enables a HTTP-based Client to communicate with the Comtarsia SignOn Gate. This solution is mainly meant for Clients, on which no Comtarsia Logon Client can be installed, e.g. Windows 95/98/ME or various UNIX derivatives. On workstations with Microsoft Windows (NT)/2000/XP and/or Linux the Comtarsia Logon Client with its extensive management possibilities can be used.

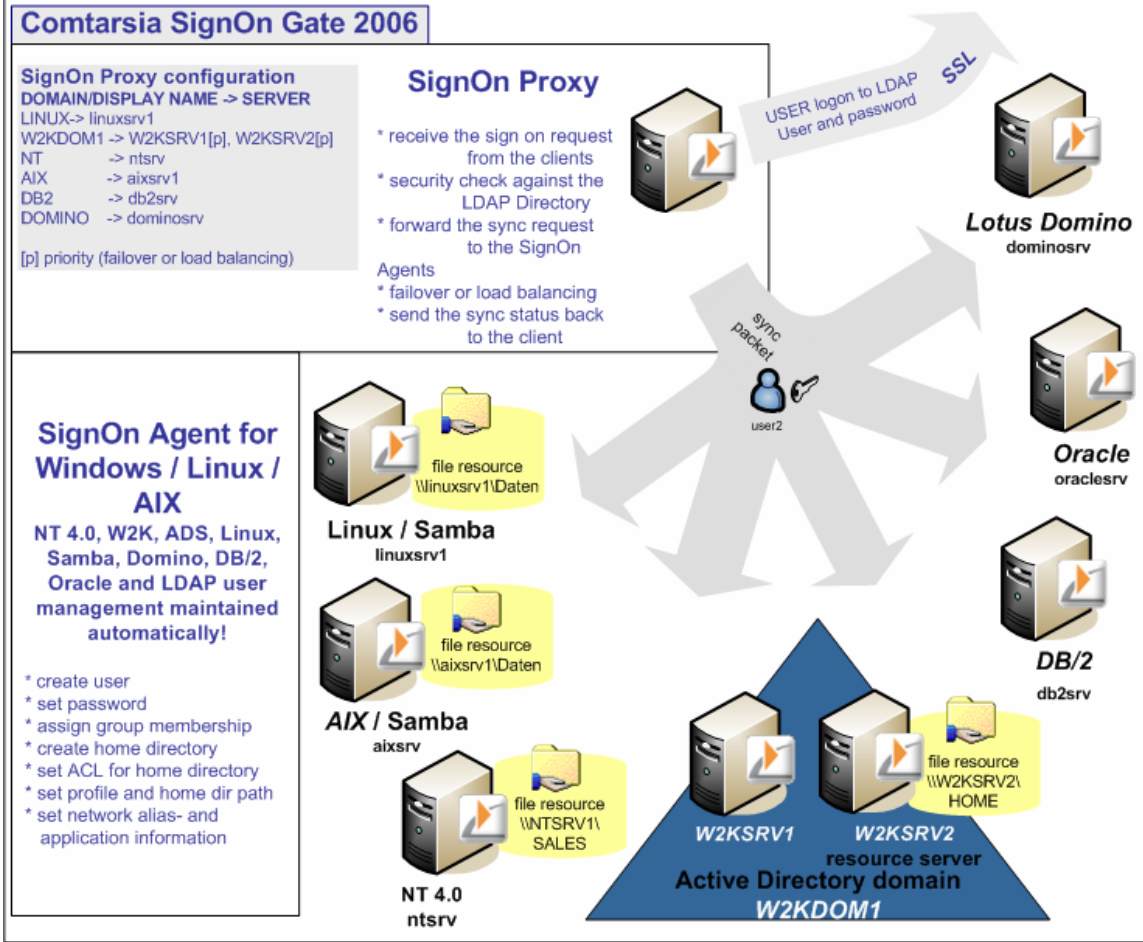
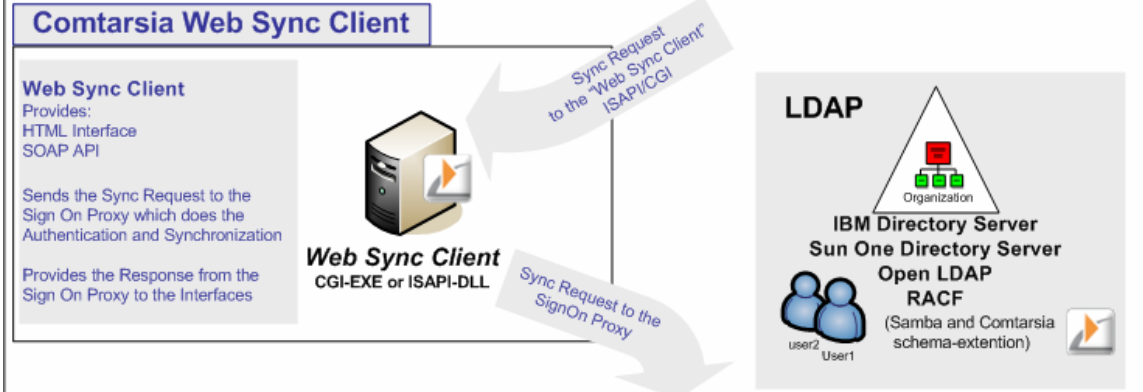
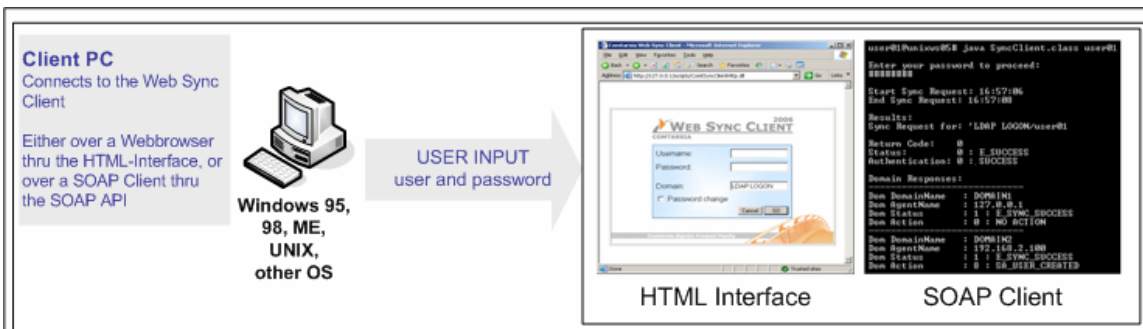
1.2 Functionality

The Comtarsia Web Sync Client makes two different operation modes available:

- 1) a web browser based HTML interface
- 2) a SOAP-API

The "Web Sync Client" transfers the "Sync Request", which it receives either over the HTML interface or over the SOAP API, to the "Sign on Proxy". The "Signon Proxy" authenticates the user against the LDAP server and sends, preconditioned a successful authentication, "Sync Requests" to all "SignOn Agents". The "SignOn Proxy" sends the status back to the "Web Sync Client".





1.2.1 HTML Interface

The final user opens the HTML-page provided by the "Web Sync Client", in a web browser (Internet Explorer, Netscape, Konqueror, Mozilla, Firefox)

By entering his logon data (user name and password) the user here has the possibility to initiate the synchronization of the logon data through the "Sign On Proxy"

Also the HTML-Interface enables the user to change his password (entering of username, old password and new password).

The HTML interface informs the user about the (un-)successful logon-synchronization process and asks the user to change his password, if it is in the "password expiration time".

1.2.2 SOAP-API

The SOAP-API provides a "software development interface" and enables therefore our customers to develop their own "Web Sync Client" interface.

A lot of program languages support SOAP (Simple Object Access Protocol), for example Java, .NET, and therefore it is secured that a SOAP-Client can be programmed for a lot of different operation systems.

SOAP Specifications

<http://www.w3.org/TR/soap/>

Apache SOAP

<http://ws.apache.org/soap/>

1.3 Installation

1.3.1 General

The following requirements are necessary:

- a web server under Windows (IIS5, IIS6)
- a server with "Comtarsia SignOn Proxy"
- a server with "Comtarsia SignOn Agent"
- a directory server (e.g. Netscape DS, IBM DS, OpenLDAP, etc.)

Of course it is possible run more than one or all of these components on only one server.

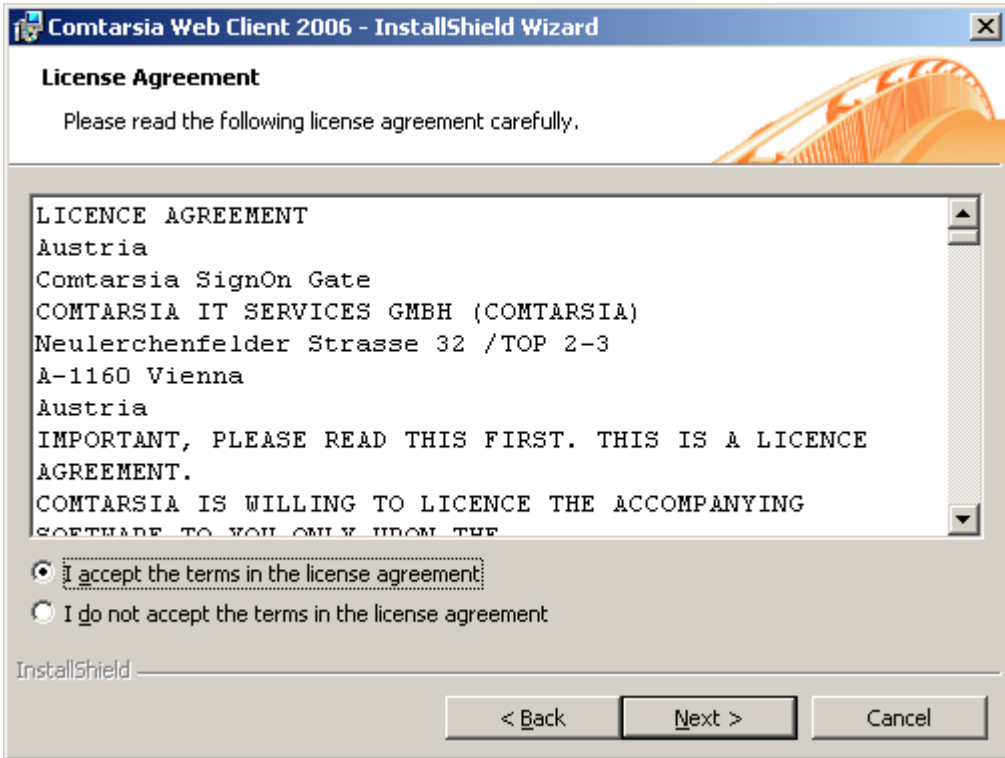
For example:

- Server1 (Windows 2000 Server): Internet Information Server (IIS), "Web Sync Client" (EXE-CGI), "Sign On Proxy", "Sign On Agent"
- Server2 (Linux): OpenLDAP

Please make sure that IIS is installed on the server.

In order to install the WebClient please start the Installer (WebClient_1.2.X.4.exe) now.





Now please put in your name and the name of your company.

Comtarsia Web Client 2006 - InstallShield Wizard

Customer Information
Please enter your information.

User Name:
comt

Organization:
comtarsia

Install this application for:

- Anyone who uses this computer (all users)
- Only for me (comt)

InstallShield

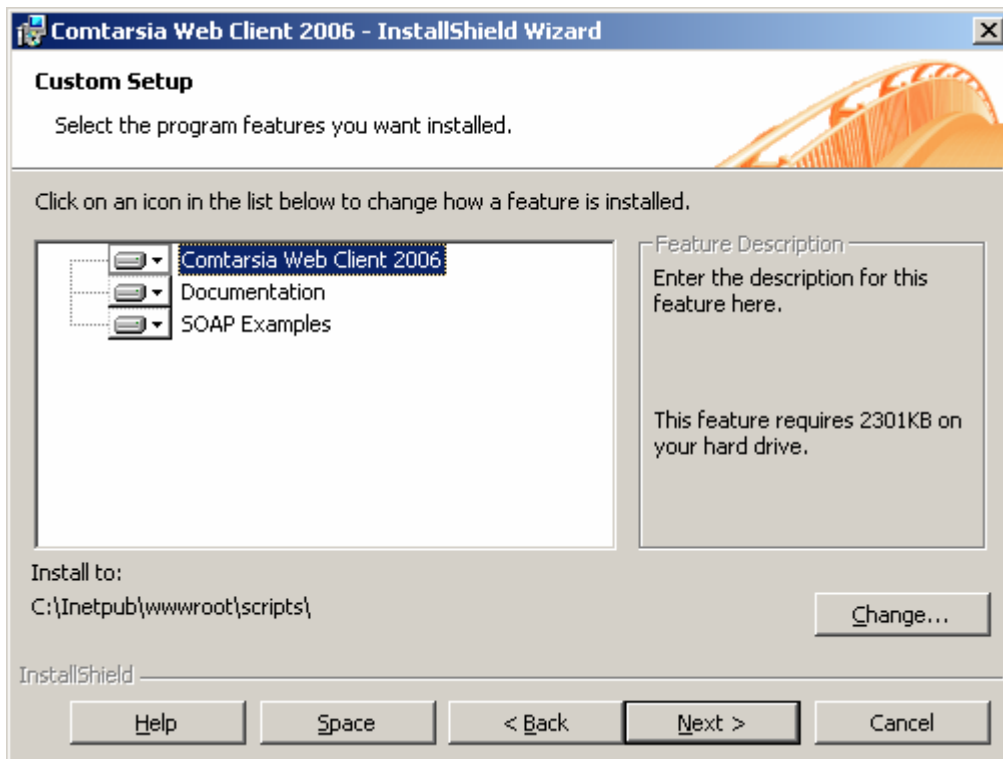
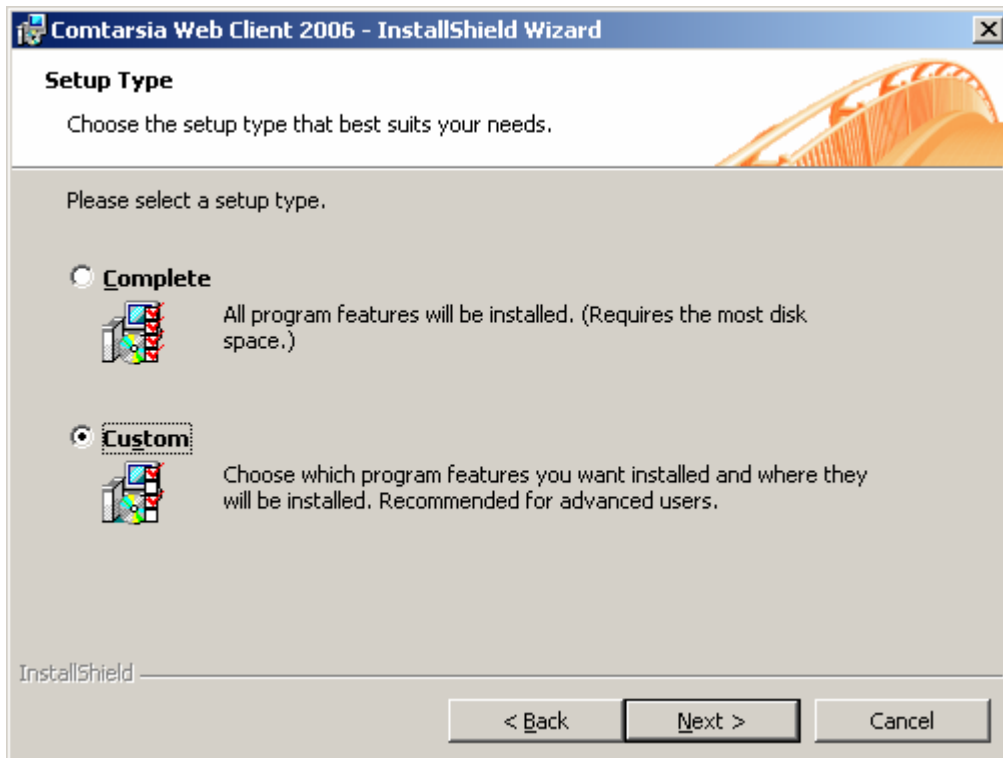
< Back Next > Cancel

Then click "Next"

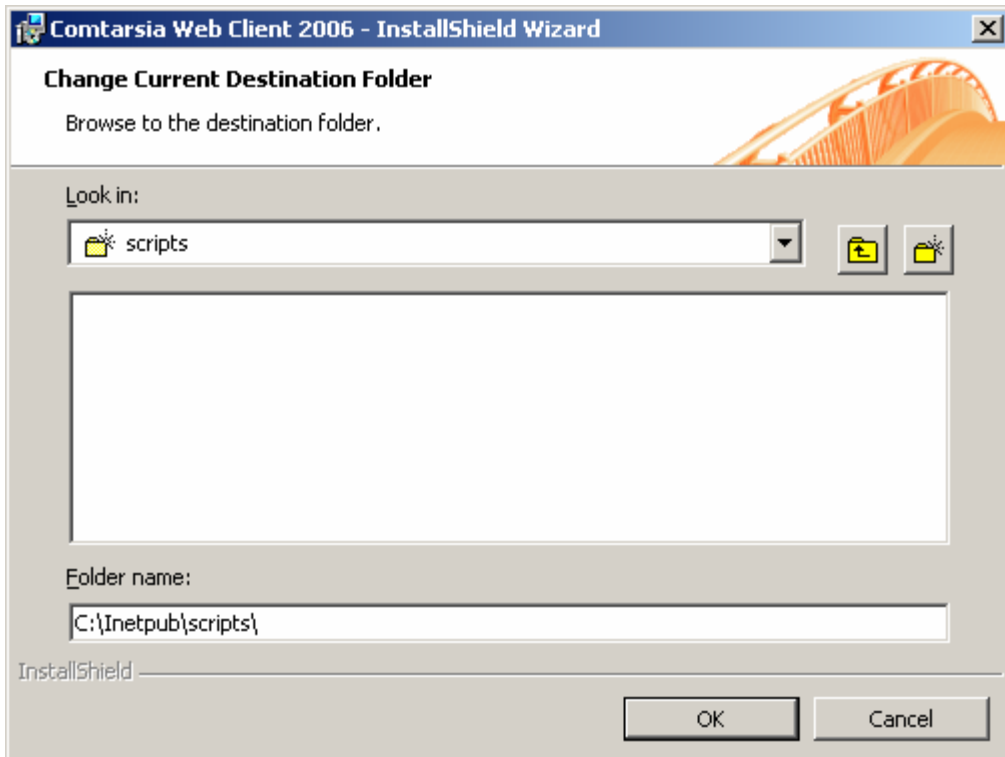
If you are using IIS5 please go on to "**Installation under IIS5**". When using IIS6 please go on to "**Installation under IIS6**".

1.3.2 Installation under IIS 5

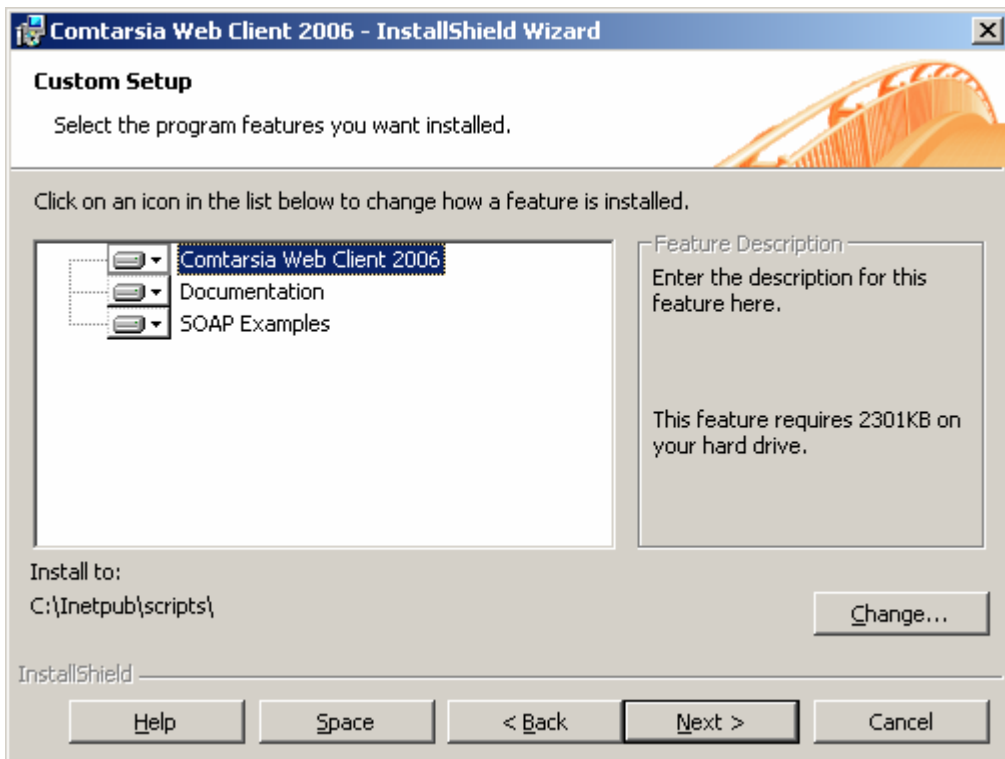
Please choose Custom as "Setup Type".



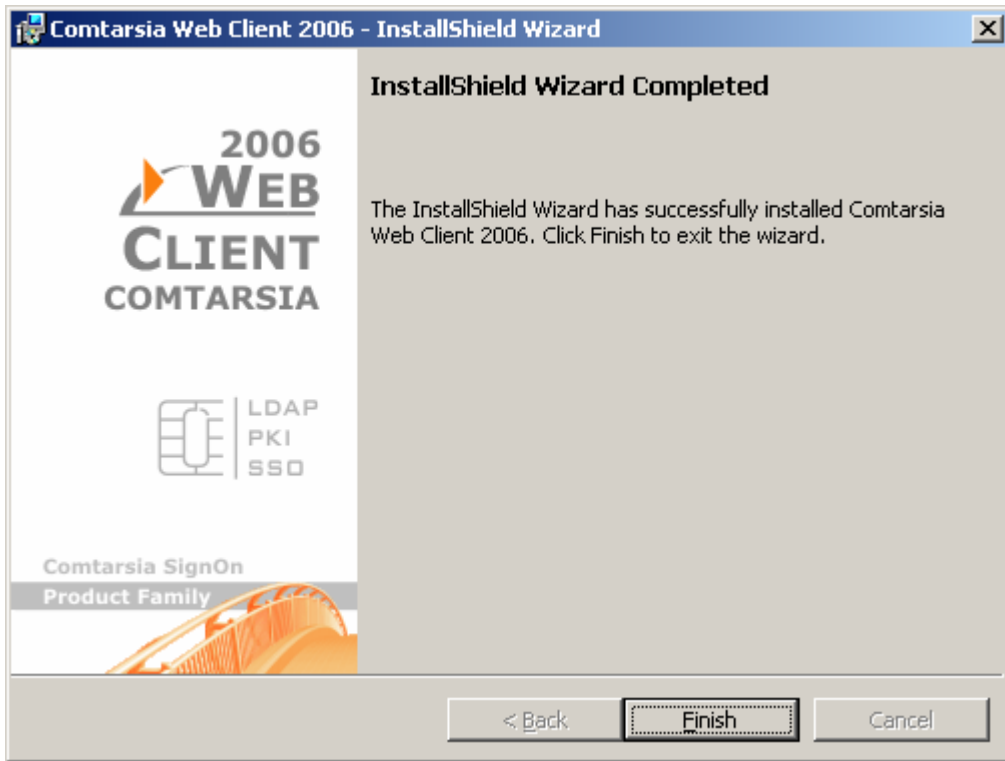
Please change the path of the "Comtarsia Web Client 2006" component to: „[IISDir]\Scripts" e.g.: „c:\InetPub\scripts\"



Confirm now by clicking "OK".



Then click "Next" and afterwards "Install".



The installation of the binary files, the documentation and the SOAP examples is now finished.

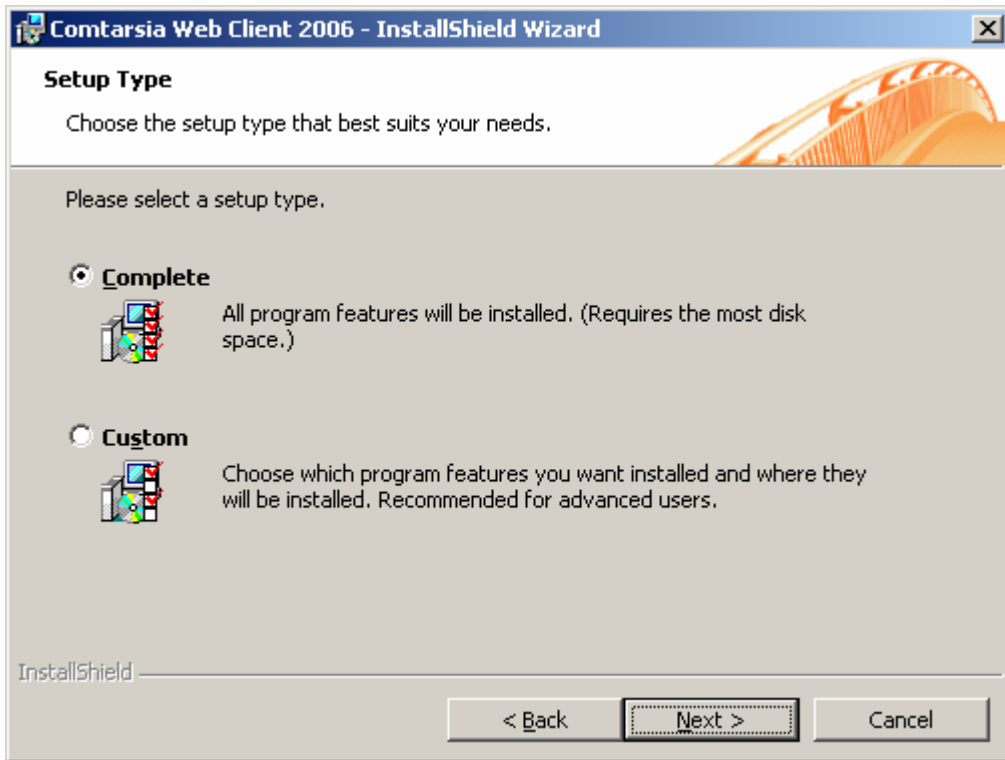
For the IIS5 no further settings have to be made.
Please go on to chapter "**Configuring the WebClient**".

1.3.3 Installation under IIS 6

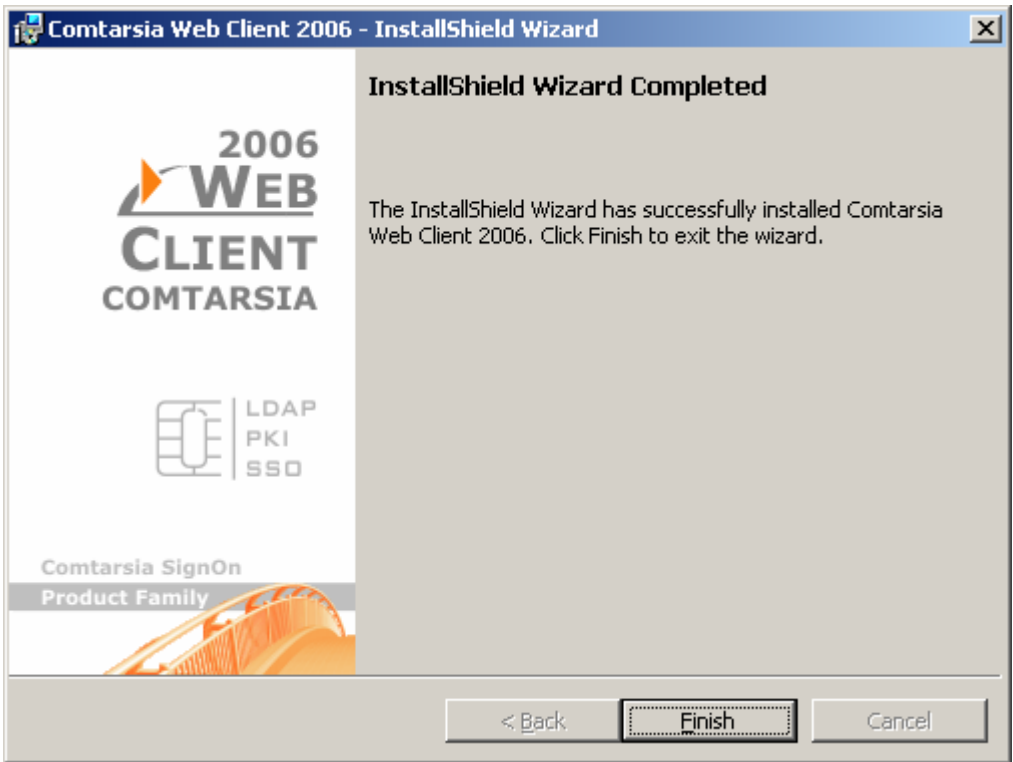
At the complete installation the WebClient, the documentation, the certificates (for the communication with the SignOn Proxy) as well as the SOAP programming examples are installed.

The WebClient is installed in the IIS-Scripts directory.

Documentation, programming examples and certificates are installed under „%programfiles%\Comtarsia\Web Client 2006\“.

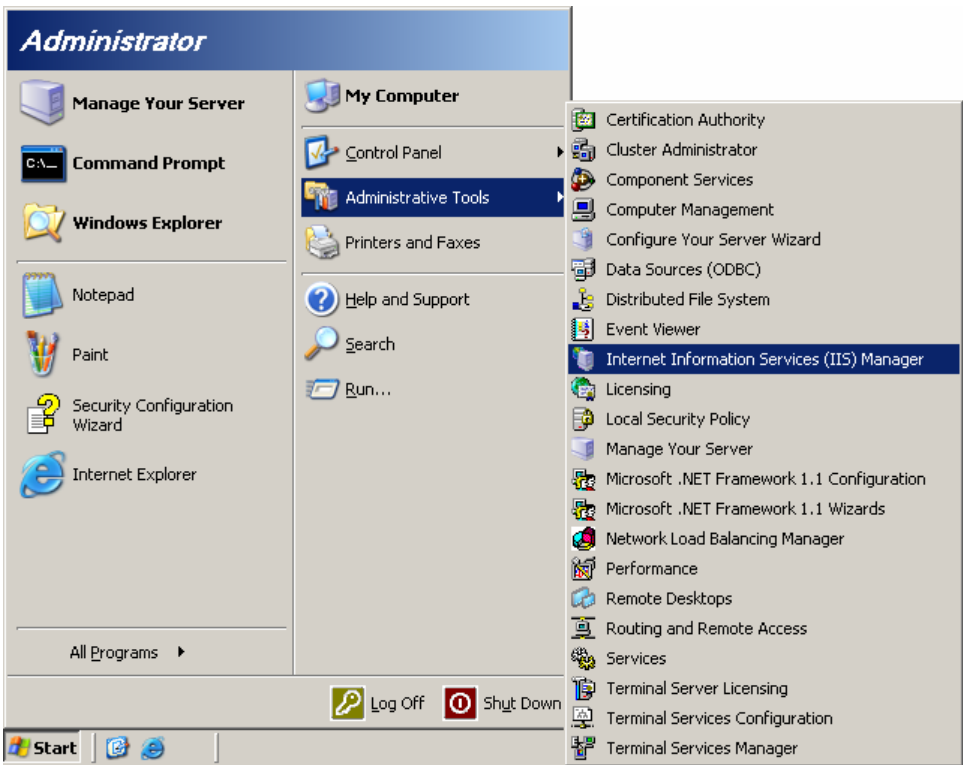


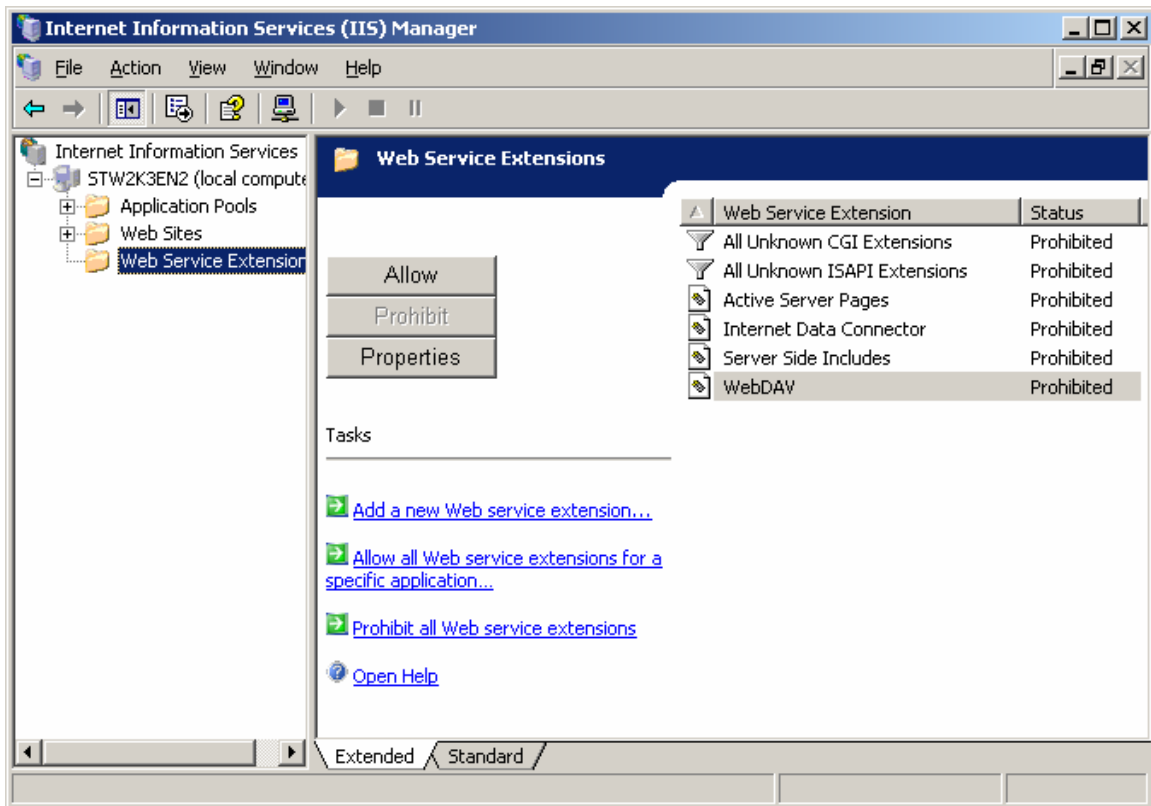
Please click "Next" and afterwards "Install".



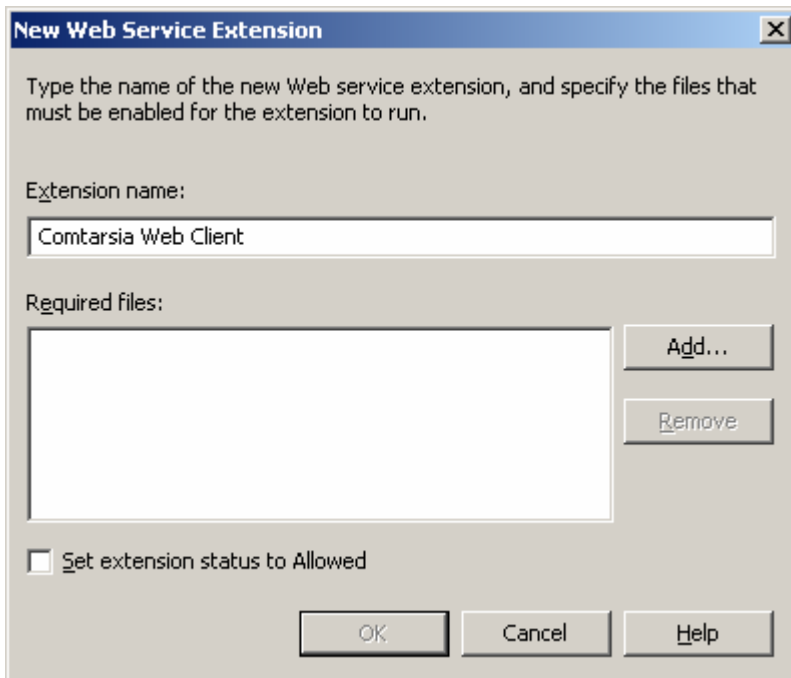
The installation of the binary files, the documentation and the SOAP examples is now finished.

Now the WebService extension for the WebClient under IIS6 has to be created. Therefore please open the "IIS management console."

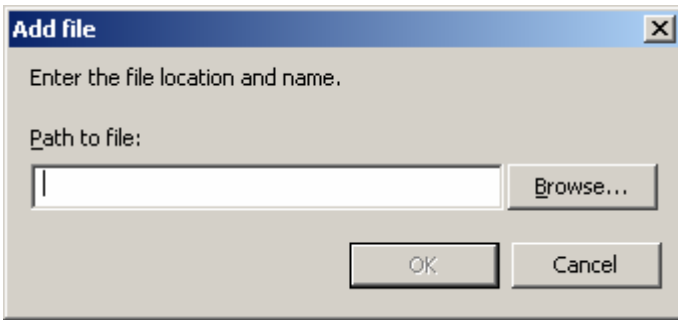




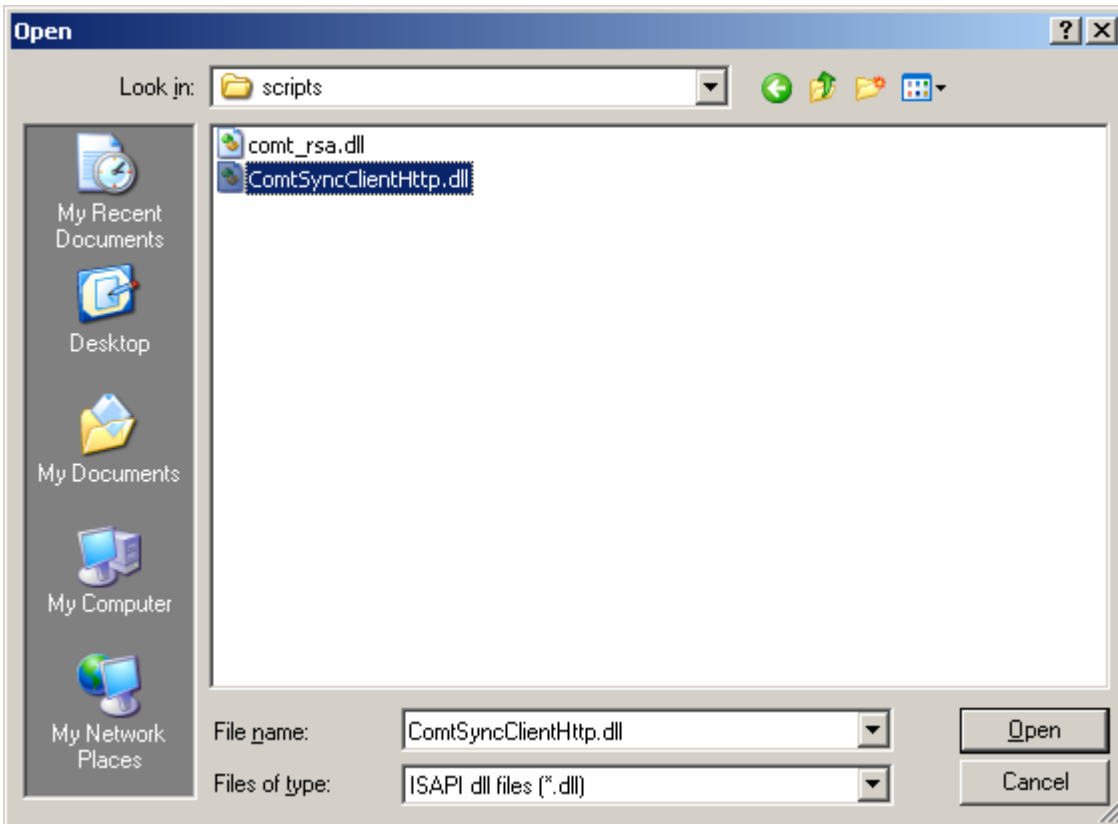
Click now “Web Service Extensions” and afterwards “Add a new Web service extension...”



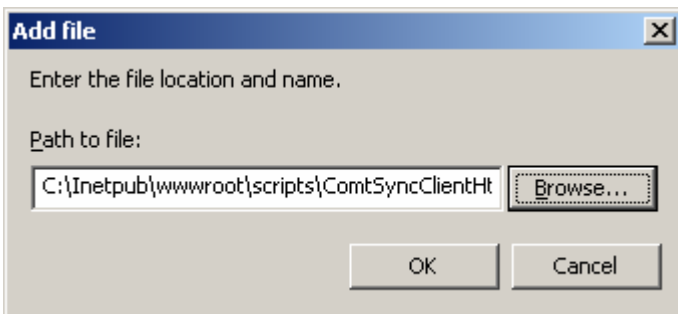
Please put in a name for the extension, for example “Comtarsia Web Client” Afterwards click “Add”.



Click "Browse" and navigate to the folder of the Comtarsia WebClient binary files in the IIS directory, e.g. „c:\inetpub\wwwroot\scripts\“

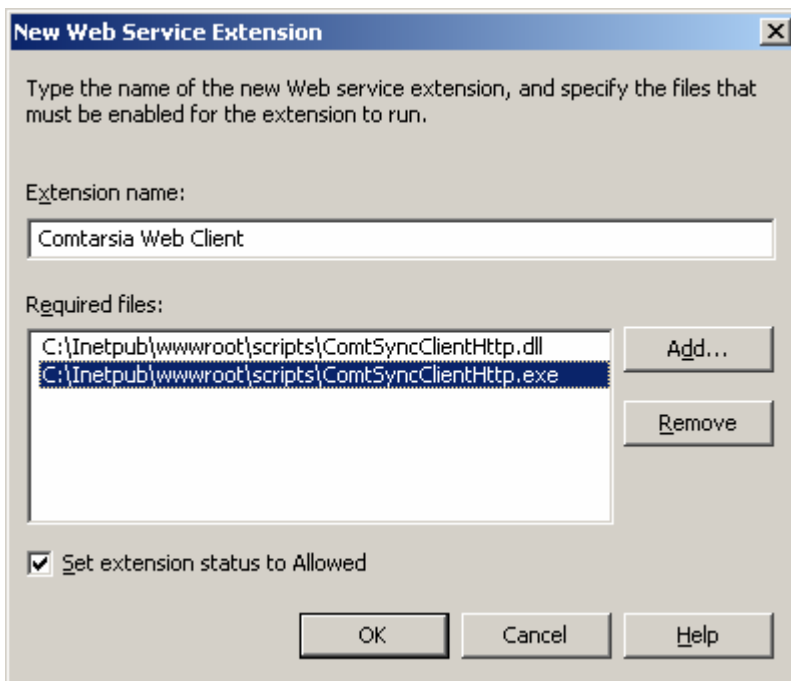
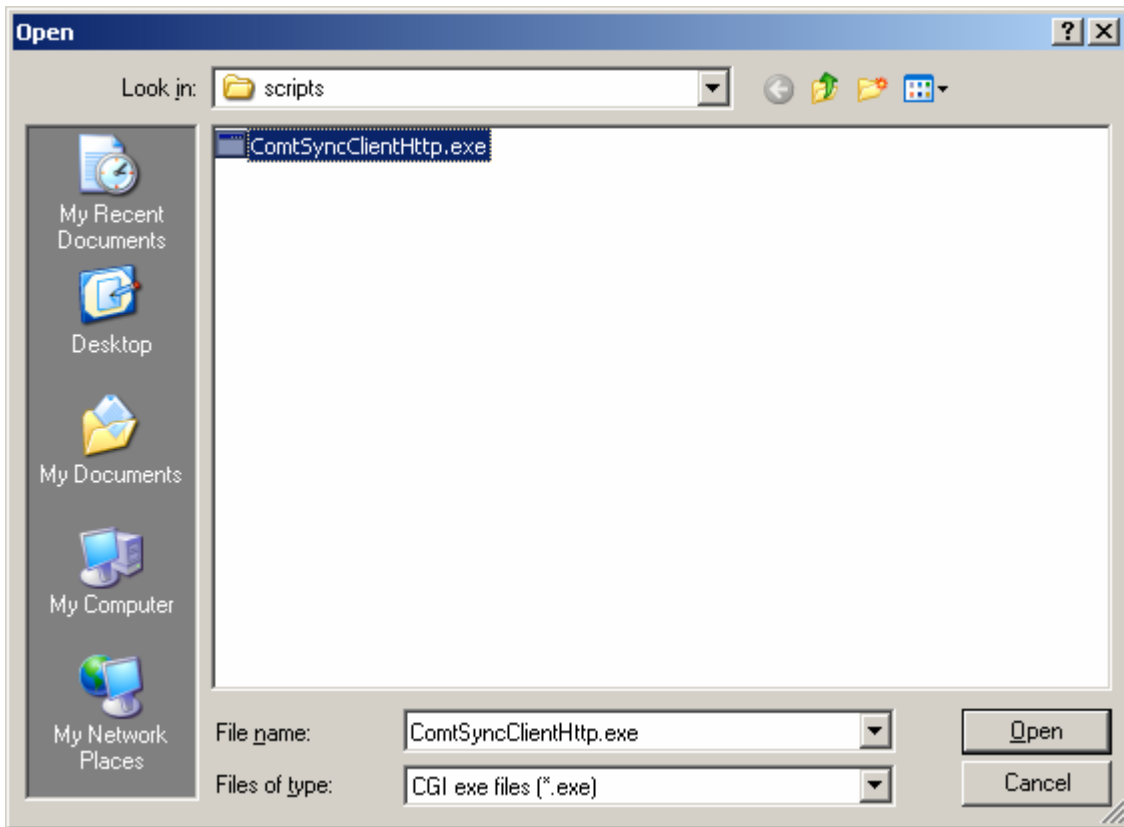


Choose the file „ComtSyncClientHttp.dll“, then click "Open" and then "OK".



Repeat the procedure for the file „ComtSyncClientHttp.exe“

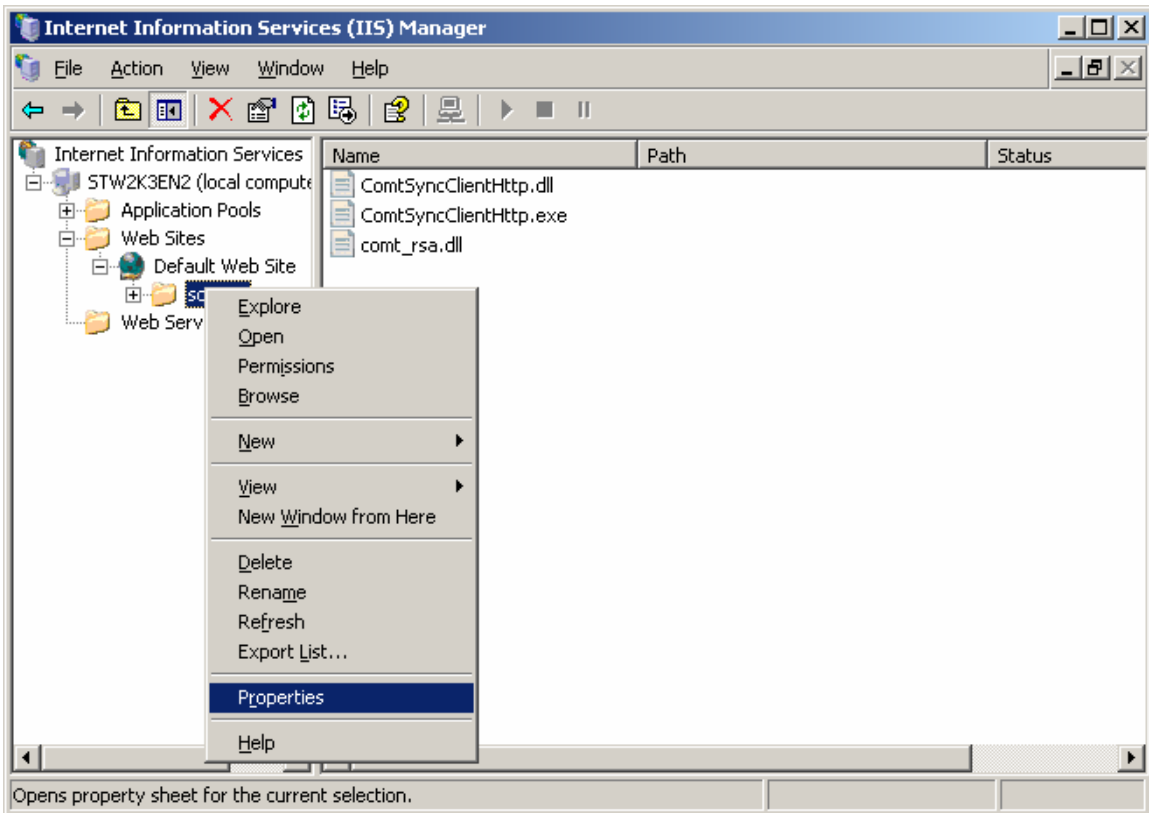




Please activate the checkbox “Set extension status to Allowed” and confirm with “OK”.

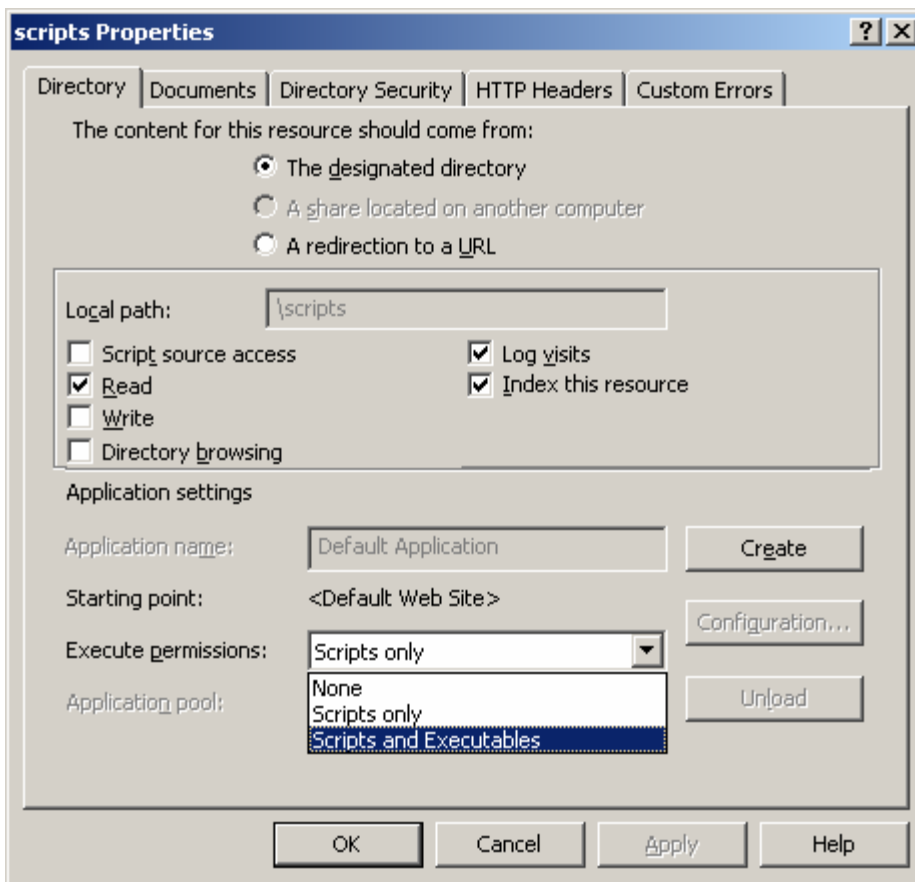


Navigate to the entry „Web Sites\Default Web Site\scripts“in the management console



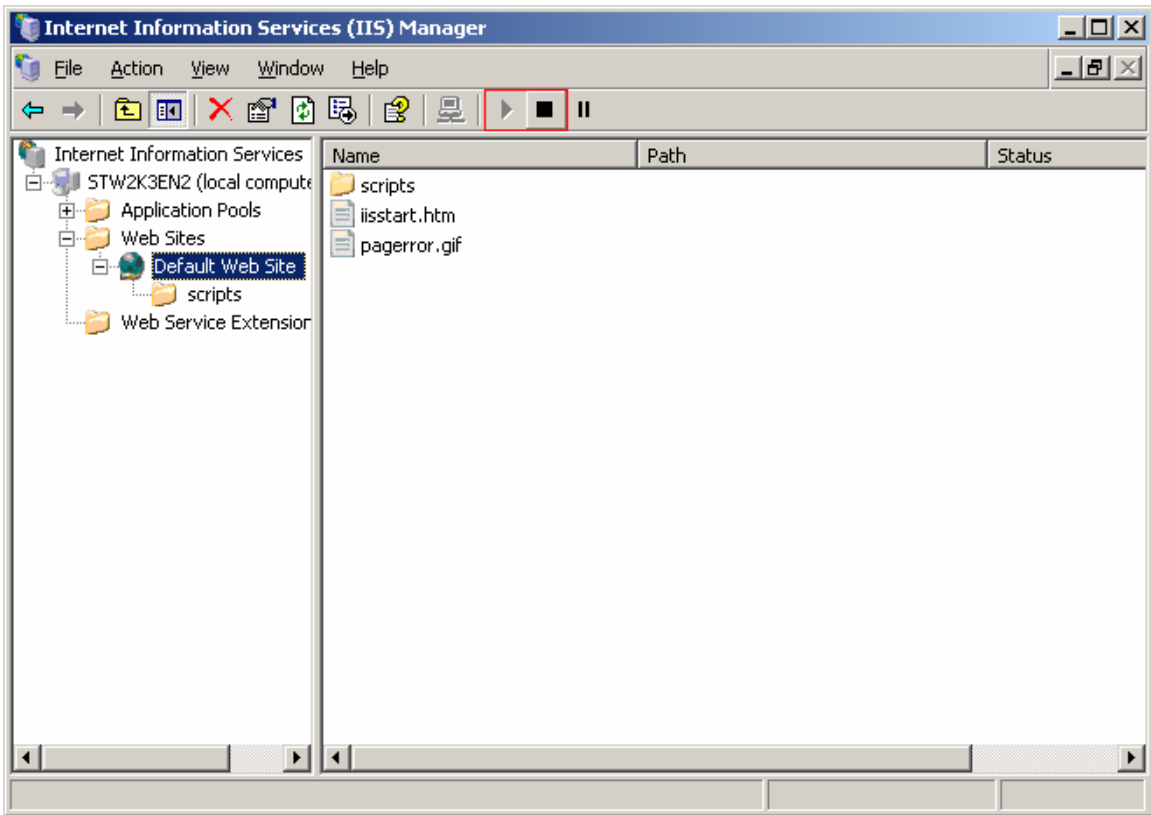
Click "Scripts" with the right mouse button and afterwards "Properties".





Change the "Execute permissions" to "Scripts and Executables".
Confirm with "Apply" and "OK".





Please start the IIS again.

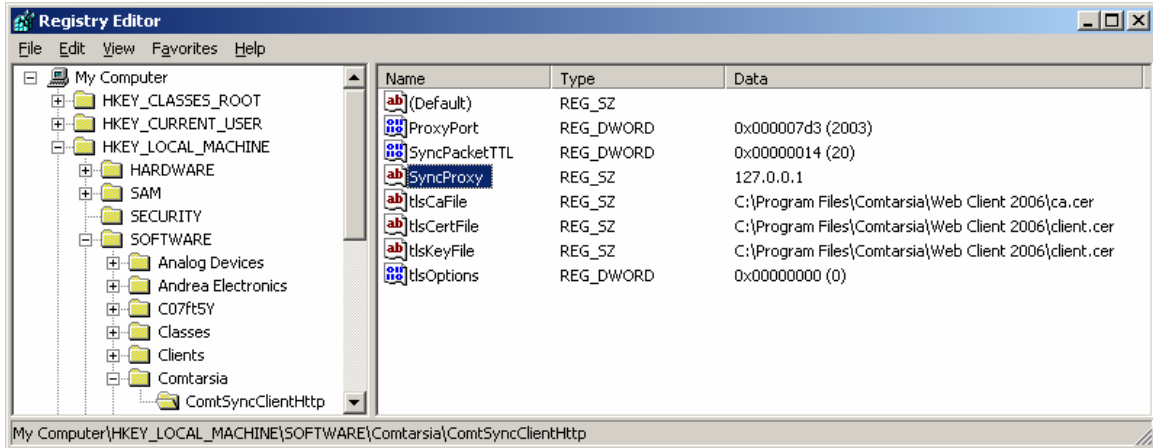
Now the installation part is finished, please go on to **“Configuring the WebClient”**.



1.3.4 Configuring the WebClient

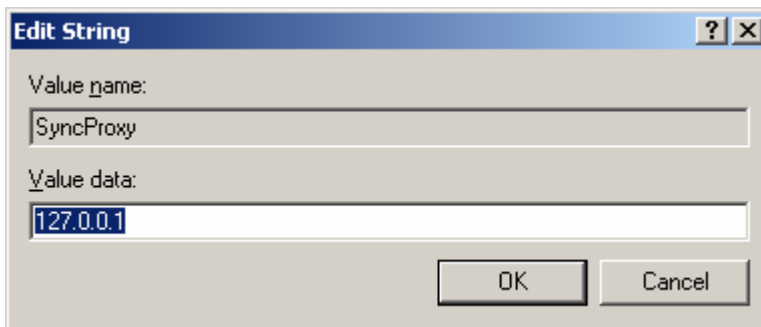
If the SignOn Proxy is installed on another than the WebServer, you have to accommodate the IP address of the Proxy server.

Therefore please execute Regedit.



Navigate to the branch

„\HKEY_LOCAL_MACHINE\Software\Comtarsia\ComtSyncClientHttp“



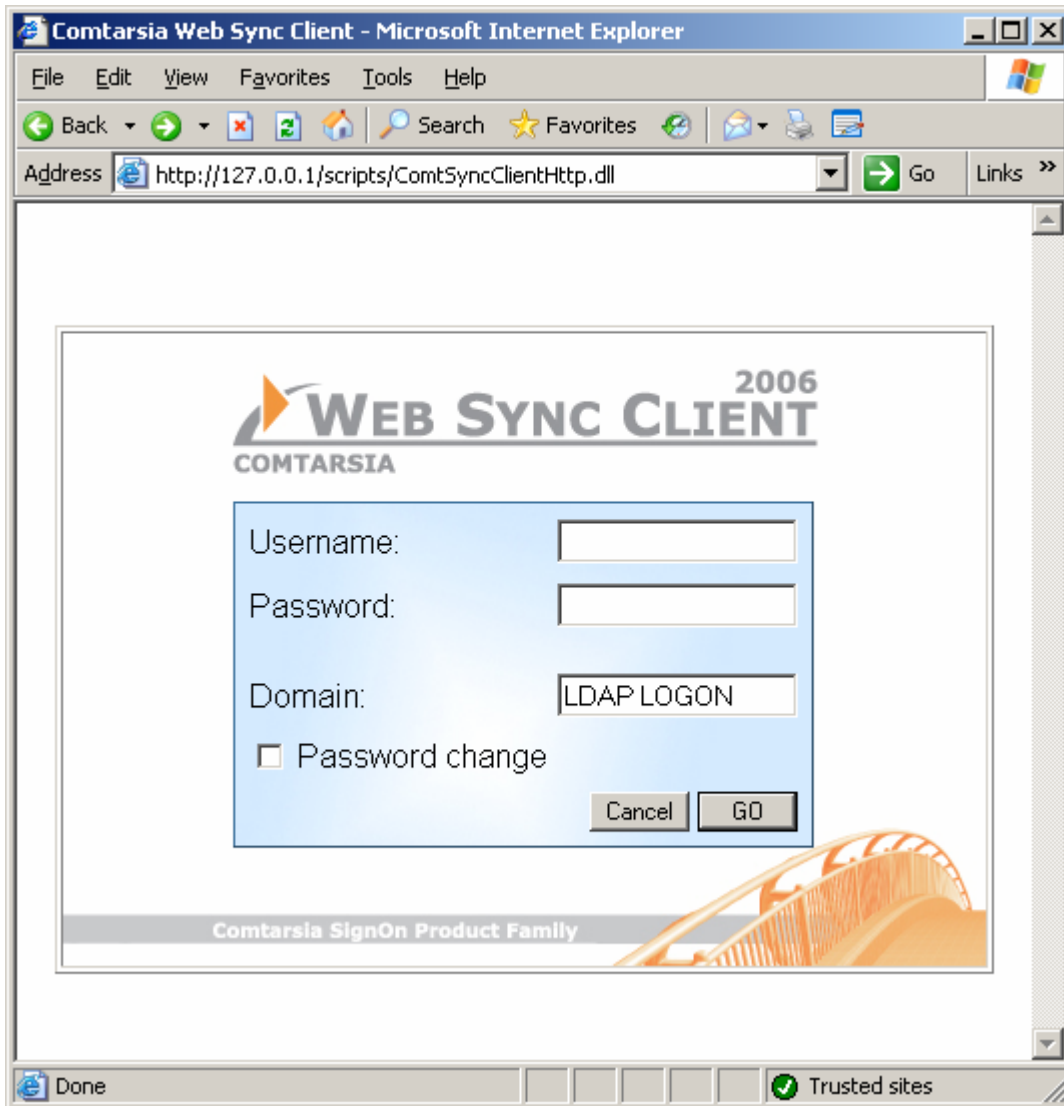
Please change the value “SyncProxy” to the IP of your Proxy server.

Congratulation! You now have successfully finished the configuration of the WebClient.

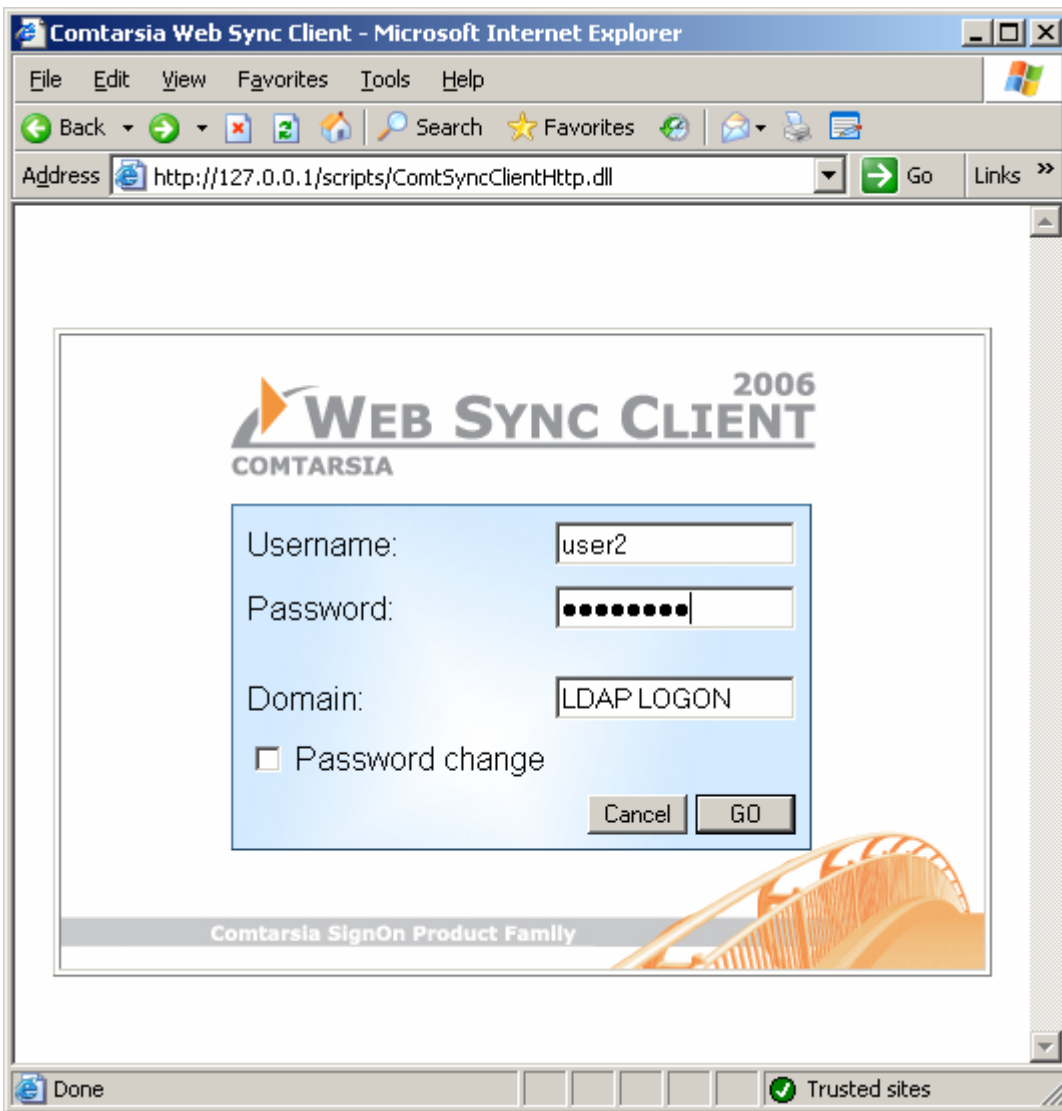
To synchronize a user via the “HTML Interface”, please go on to “**The HTML Interface**”.

1.3.5 The HTML Interface

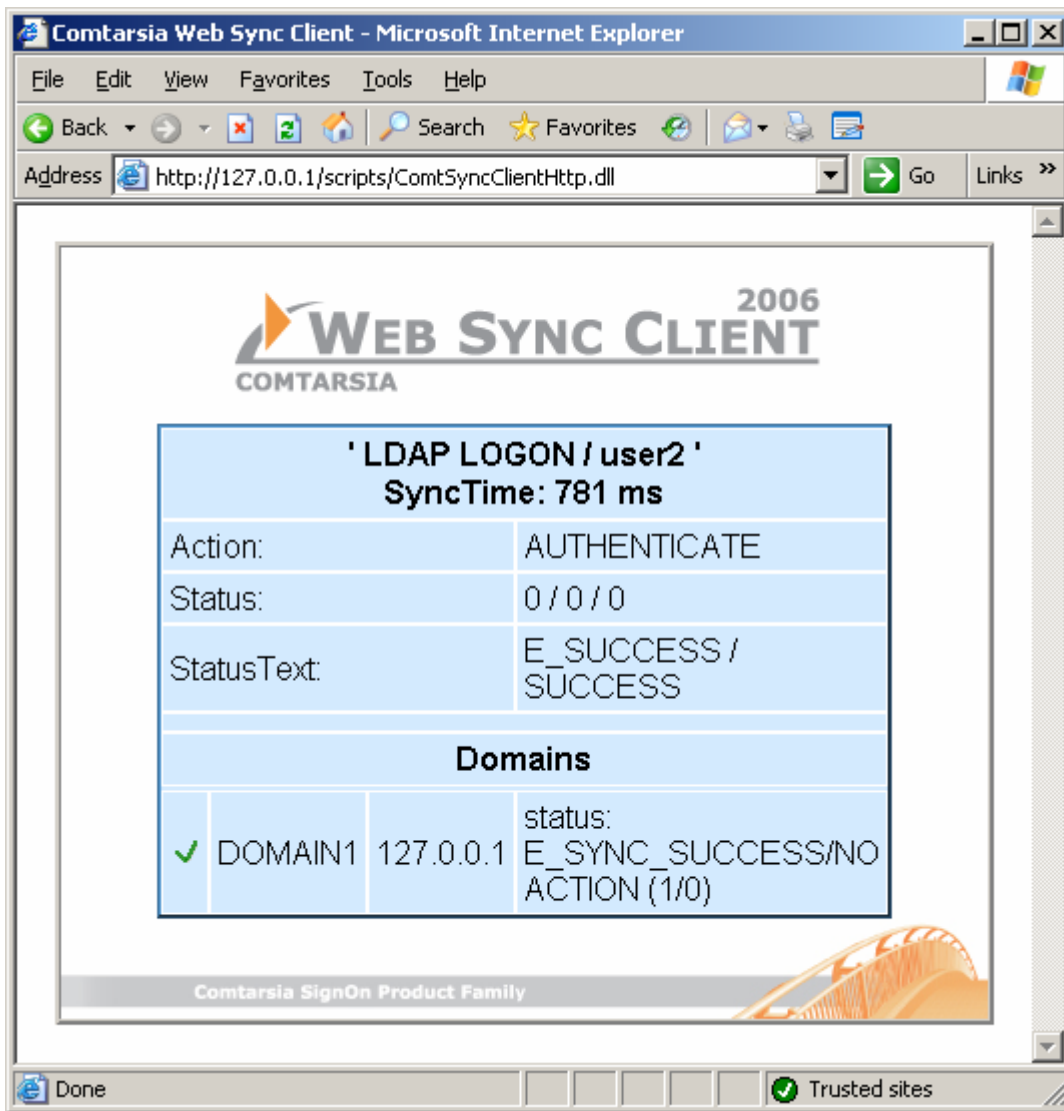
To configure the HTML-Interface there are no further steps necessary.
To use the HTML-Interface simply navigate with a web browser of your choice to the following address: `http://<yourservname>/scripts/ComtSyncClientHttp.exe` or `http://<yourservname>/scripts/ComtSyncClientHttp.dll`



To synchronize a user simply put in username and password of the LDAP user object and confirm with "OK".



[Input of username and password]



[Answer of a successful synchronization]

1.3.6 SOAP-API

The file "ComtSyncClientHttp.exe" or "ComtSyncClientHttp.dll" simultaneously works as SOAP-Server. Therefore no further configurations are necessary.

WSDL:

http://<name\hresServers>/scripts/ComtSyncClientHttp.[dll|exe]?WSDL

For further examples on developing a SOAP Client please see the following chapter "[Examples for SOAP Client-Development](#)"

1.4 Examples for SOAP Client-Development

1.4.1 PHP SOAP-Client

In order to develop a PHP-SOAP Client the following is required:
a web server (e.g. IIS6, Apache, etc.)
PHP5 (www.php.net)

Windows:

PHP 5.0.4 zip package
(the extensions are not included in the Installer, but the SOAP Extension is required)

Linux:

PHP 5.0.4 source
Create php with a SOAP extension.

Please change the configuration of the Http-Web server, so that PHP-Scripts are processed through the PHP Interpreter (see install.txt → is delivered with PHP)

Please insert the following line into the PHP.ini in order to activate the SOAP-Extension

under Windows

```
extension=php_soap.dll
```

under Linux:

```
extension=php_soap.so
```

Please note also that the parameter "extension_dir" is pointed to the directory with the extensions.

For example:

```
extension_dir = "./"
```

or

```
extension_dir = "c:/php/ext/"
```

Now start the web server again.

Then copy our delivered "*.php" files into a directory of your web server.

For example

```
c:\apache\apache2\htdocs\comtphp\
```

To check if PHP, including the SOAP Extension, works on your web server properly please navigate with a web browser to the former copied phpinfo.php file.

For example:

```
http://server.comtarsia.com/comtphp/phpinfo.php
```

If all this worked well, you now see a PHP information site.

Further down this site all activated PHP Extensions are listed. Now the SOAP Extension should also be listed.

Now please edit the following files:

```
phpsoap.php, phpsoap2.php
```



so that the variable \$wsdlUrl points to your wsdl-file

for example

```
$wsdlUrl="http://server.comtarsia.com/ComtarsiaWebSyncClient_CGI.wsdl";
```

In the file phpsoap.php you have to insert additionally a user name and a password for testing purposes.

Now you can navigate to one of these files with the web browser to test the connection to the "Web Sync Client" via the SOAP using PHP.

Then you can edit the files as you like and can adjust them to your convenience.

1.4.2 Java SOAP-Client

To create a Java SOAP Client the following is required:

Sun: Java2SE JDK (Java Development Kit)

(www.sun.com)

["http://www.sun.com/download/index.jsp?cat=Application%20Development&tab=3&subcat=SDKs%20\(Software%20Development%20Kits\)"](http://www.sun.com/download/index.jsp?cat=Application%20Development&tab=3&subcat=SDKs%20(Software%20Development%20Kits))
java.sun.com/j2se/downloads

JavaMail (mail.jar) included in J2EE

<http://java.sun.com/products/javamail/downloads/index.html>

JavaBeans Activation Framework (activation.jar) included in J2EE

<http://java.sun.com/products/javabeans/glasgow/jaf.html>

Apache WebServices - Axis (axis-bin-1_2RC3.zip)

http://www.apache.org/dyn/closer.cgi/ws/axis/1_2RC3

(All ".jar" files out of axis-1_2RC3\lib")

Please install the JDK first.

Afterwards copy the downloaded "jar" files into the following directory

[JDKInstallation]\jre\lib\ext

therefore the APIs are available.

for example: c:\Program Files\java\jdk1.5.0_02\jre\lib\ext

If you also install the JRE (Java Runtime Environment), which is delivered with the

JDK, you also have to copy the "jar" file to

[JREInstallation]\lib\ext

for example: c:\Program Files\java\jre1.5.0_02\lib\ext

Now please unpack the delivered JavaSOAP.zip file in a directory of your choice (e.g.

c:\devel\ComtSoap\)

To create a connection to the SOAP server please open the command prompt and change into the development directory of the SOAP Client.

For example: c:\devel\ComtSoap



Then create the necessary classes via the command line using the WSDL2Java.

```
java org.apache.axis.wsdl.WSDL2Java (WSDL-file-URL)
The parameter (WSDL-file-URL) has to point to your web server.
```

For example:

```
C:\devel\ComtSoap>java org.apache.axis.wsdl.WSDL2Java -v -a -D -W
http://server.comtarsia.com/ComtarsiaWebSyncClient_CGI.wsdl
```

In the directory "com\comtarsia\signon\WebSyncClient" the necessary classes are created automatically.

Optionally you can use the batch file "createClasses.bat " (the file has to be in the development directory)

For this purpose change in the file the following line
set WSDLPath=http://server.comtarsia.com/ComtarsiaWebSyncClient_CGI.wsdl
to the path of your wsdl file.

for example:

```
set WSDLPath=http://myserver.mycompany.com/mywsdl.wsdl
```

To create the examples simply execute "buildSamples.bat".

Eventually you will have to adjust the file, so that the "Java2SEPath" variable points to your Java2SE installation path.

Afterwards you can execute the examples using

```
java -cp "." comtSOAP username password
```

and

```
java -cp "." comtSOAP2
```

The example comtSOAP is set up so that you deliver username and password as parameter.

You can adjust the source code of the "ComtSOAP.java" and the "ComtSOAP2.java" to your convenience.

To compile the files again, simply execute "buildSamples.bat" repeatedly.

To execute only the SOAP Client (e.g. on just one Client-PC) the following is required:

Sun: Java2RE

(www.sun.com) java.sun.com/j2se/downloads

(Note: In comparison with SDK (JDK) the JRE (Java Runtime Environment) requires less disk space.)

And the "jar" files of the following packages:

JavaMail (mail.jar)

<http://java.sun.com/products/javamail/downloads/index.html>

JavaBeans Activation Framework (activation.jar)

<http://java.sun.com/products/javabeans/glasgow/jaf.html>

Apache WebServices - Axis

http://www.apache.org/dyn/closer.cgi/ws/axis/1_2RC3



The downloaded "jar" files are copied into the following directory

[J2SEInstallation]\lib\ext

Therefore the APIs are available.



Call parameter:

```
ComtarsiaWebSyncClient cwsc = new ComtarsiaWebSyncClient();
SyncResponse_1 sr = cwsc.doSync_1(USERNAME, PASSWORD, NEWPASSWORD,
LOGON_DOMAIN, ACTION);
```

```
#define ACTION_AUTHENTICATE 0
#define ACTION_PWD_CHANGE 1
```

Return values for "sr.status":

```
#define E_SUCCESS 0x0
#define E_ILLEGAL_REQUEST 0x10
#define E_SYNC_TIMEOUT 0x11
#define E_SYNC_USER 0x12
#define E_AUTHENTICATION 0x13
#define E_COMMUNICATION 0x14
```

// LastErr ERRORCODES return value is always E_RETVAL_SUCCESS

```
#define E_INTERNAL 0x200
#define E_NETWORK 0x100
```

```
#define E_LASTERR_COMT_RSA_VERSION (E_INTERNAL + 1) // Wrong
COMT_RSA version
#define E_LASTERR_REGISTRY (E_INTERNAL + 2) // Error
reading registry values
#define E_LASTERR_UNKNOWN_FLAG_VAL (E_INTERNAL + 3) // Unknown
Smem FLAG value
#define E_LASTERR_WSASTARTUP (E_INTERNAL + 4) //
Wsastartup problem
#define E_LASTERR_ENCRYPTION_TYPE (E_INTERNAL + 5) // Proxy and
ComtSyncClient encryption types do not match
#define E_LASTERR_RSA_AQUIRE_CTX (E_INTERNAL + 6) // RSA
acquire context error
#define E_LASTERR_KEY (E_INTERNAL + 7) // Some
error with an RSA KEY ocured

#define E_LASTERR_RESOLVING_PROXY (E_NETWORK + 1) // PROXY
name can not be resolved
#define E_LASTERR_CONNECT (E_NETWORK + 2) // Connect
problem to PROXY
#define E_LASTERR_RECEIVE (E_NETWORK + 3) // Receive
error
#define E_LASTERR_SEND (E_NETWORK + 4) // Send
error
#define E_LASTERR_SOCKET_CREATION (E_NETWORK + 5) // Socket
creation problem
#define E_LASTERR_ILLEGAL_MSG_HEADER (E_NETWORK + 6) // Header
check failed illegal or not expected message header
```



Return values for "sr.statusAuth":

```
/* comt_ldap return values */
#define COMT_LDAP_RC_SUCCESS 0x00
#define COMT_LDAP_RC_ERROR 0x01
#define COMT_LDAP_RC_WRONG_PWD 0x02
#define COMT_LDAP_RC_PWD_EXPIRED 0x03
#define COMT_LDAP_RC_PWD_POLICY_VIOLATION 0x04
#define COMT_LDAP_RC_USERID_REVOKED 0x05
#define COMT_LDAP_RC_USERID_UNDEFINED 0x06
#define COMT_LDAP_RC_NO_SUCH_OBJECT 0x07
#define COMT_LDAP_RC_WRONG_VERSION 0x11
#define COMT_LDAP_RC_MISSING_DATA 0x12
#define COMT_LDAP_RC_UNSUPPORTED 0x13
/* comt_ldap communication error return code */
#define COMT_LDAP_RC_COM_ERROR 0x1a
/* ietf error codes and warnings */
#define COMT_LDAP_RC_IETF_WARN_EXPIRE 0x20
#define COMT_LDAP_RC_IETF_WARN_GRACE 0x21
/* ietf error codes have to start at offset COMT_LDAP_RC_IETF_ERROR_START
   because they are directly mapped */
#define COMT_LDAP_RC_IETF_ERROR_START 0x30
#define COMT_LDAP_RC_IETF_ERROR_EXPIRED 0x30
#define COMT_LDAP_RC_IETF_ERROR_ACC_LOCKED 0x31
#define COMT_LDAP_RC_IETF_ERROR_CHG_RESET 0x32
#define COMT_LDAP_RC_IETF_ERROR_NO_MOD 0x33
#define COMT_LDAP_RC_IETF_ERROR_NEED_OLD_PWD 0x34
#define COMT_LDAP_RC_IETF_ERROR_INVALID_SYNTAX 0x35
#define COMT_LDAP_RC_IETF_ERROR_PWD_TOO_SHORT 0x36
#define COMT_LDAP_RC_IETF_ERROR_PWD_TOO_YOUNG 0x37
#define COMT_LDAP_RC_IETF_ERROR_PWD_IN_HISTORY 0x38
```



Return values for „sr.domains[i].status“:

#define E_SYNC_SUCCESS	0x1
#define E_SYNC_ERROR	0x2
#define E_NO_SYNC_AGENT	0x4
#define E_SP_PROXY_TIMEOUT	0x8
#define E_SP_AGENT_TIMEOUT	0x10
#define E_SP_AUTHENTICATION	0x20

Return values for „sr.domains[i].action“:

#define SA_USER_ENABLED	0x1
#define SA_USER_PW_SET	0x2
#define SA_USER_DELETED	0x4
#define SA_USER_CREATED	0x8
#define SA_USER_GRP_ADDED	0x10
#define SA_USER_GRP_DELETED	0x20
#define SA_USER_OU_MOVE	0x40
#define SA_USER_INFO_UPDATED	0x80



```

<?xml version="1.0" encoding="utf-8" ?>
: <wsdl:definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://signon.comtarsia.com/WebSyncClient/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  targetNamespace="http://signon.comtarsia.com/WebSyncClient/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
: <wsdl:types>
: <s:schema elementFormDefault="qualified"
  targetNamespace="http://signon.comtarsia.com/WebSyncClient/">
: <s:element name="doSync_1Request">
: <s:complexType>
: <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="username" type="s:string" />
  <s:element minOccurs="1" maxOccurs="1" name="password" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="passwordNew" type="s:string" />
  />
  <s:element minOccurs="1" maxOccurs="1" name="domain" type="s:string" />
  <s:element minOccurs="1" maxOccurs="1" name="action" type="s:int" />
  </s:sequence>
  </s:complexType>
  </s:element>
: <s:element name="doSync_1Response">
: <s:complexType>
: <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="doSync_1Result"
    type="tns:SyncResponse_1" />
  </s:sequence>
  </s:complexType>
  </s:element>
: <s:complexType name="SyncResponse_1">
: <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="rc" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="status" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="statusText" type="s:string" />
  <s:element minOccurs="1" maxOccurs="1" name="statusAuth" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="statusAuthText" type="s:string" />
  />
  <s:element minOccurs="0" maxOccurs="1" name="domains"
    type="tns:ArrayOfDomainResponse_1" />
  </s:sequence>
  </s:complexType>
: <s:complexType name="ArrayOfDomainResponse_1">
: <s:sequence>
  <s:element minOccurs="0" maxOccurs="unbounded" name="DomainResponse_1"
    nillable="true" type="tns:DomainResponse_1" />
  </s:sequence>
  </s:complexType>
: <s:complexType name="DomainResponse_1">

```



```

: <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="domainName" type="s:string"
    />
  <s:element minOccurs="1" maxOccurs="1" name="agentName" type="s:string" />
  <s:element minOccurs="1" maxOccurs="1" name="status" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="statusText" type="s:string" />
  <s:element minOccurs="1" maxOccurs="1" name="action" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="actionText" type="s:string" />
</s:sequence>
</s:complexType>
</s:schema>
</wsdl:types>
: <wsdl:message name="doSync_1SoapIn">
  <wsdl:part name="parameters" element="tns:doSync_1Request" />
</wsdl:message>
: <wsdl:message name="doSync_1SoapOut">
  <wsdl:part name="parameters" element="tns:doSync_1Response" />
</wsdl:message>
: <wsdl:portType name="ComtarsiaWebSyncClientSoap">
: <wsdl:operation name="doSync_1">
  <wsdl:input message="tns:doSync_1SoapIn" />
  <wsdl:output message="tns:doSync_1SoapOut" />
</wsdl:operation>
</wsdl:portType>
: <wsdl:binding name="ComtarsiaWebSyncClientSoap"
  type="tns:ComtarsiaWebSyncClientSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
    style="document" />
: <wsdl:operation name="doSync_1">
  <soap:operation
    soapAction="http://signon.comtarsia.com/WebSyncClient/doSync_1"
    style="document" />
: <wsdl:input>
  <soap:body use="literal" />
</wsdl:input>
: <wsdl:output>
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
: <wsdl:service name="ComtarsiaWebSyncClient">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/" />
: <wsdl:port name="ComtarsiaWebSyncClientSoap"
  binding="tns:ComtarsiaWebSyncClientSoap">
  <soap:address
    location="HTTP://192.168.2.75:80/scripts/ComtSyncClientHttp.exe" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

