



Comtarsia Web Client 2006

Einführung und Installation

Version: 1.2.5.4, 04-Jul-2006

Inhaltsverzeichnis

1.	Comtarsia Web Client	3
1.1	Einführung	3
1.2	Funktionsbeschreibung	3
1.2.1	HTML Interface	5
1.2.2	SOAP-API	5
1.3	Installation	5
1.3.1	Generell	5
1.3.2	Installation unter IIS 5	8
1.3.3	Installation unter IIS 6	13
1.3.4	Konfiguration des WebClient	22
1.3.5	Das HTML Interface	23
1.3.6	SOAP API	25
1.4	SOAP Client Entwicklungs-Beispiele	26
1.4.1	PHP SOAP-Client	26
1.4.2	Java SOAP-Client	27



1. Comtarsia Web Client

1.1 Einführung

Der Comtarsia Web Sync Client stellt einen HTTP-basierenden Client zur Kommunikation mit dem Comtarsia SignOn Gate zur Verfügung. Diese Lösung ist hauptsächlich für Clients gedacht, auf welchen kein Comtarsia Logon Client installiert werden kann, wie z.B. Windows 95/98/ME oder diverse UNIX-Derivate. Auf Arbeitsplätzen mit Microsoft Windows (NT)/2000/XP bzw. Linux kann der Comtarsia Logon Client mit seinen umfangreichen Management-Möglichkeiten zum Einsatz kommen.

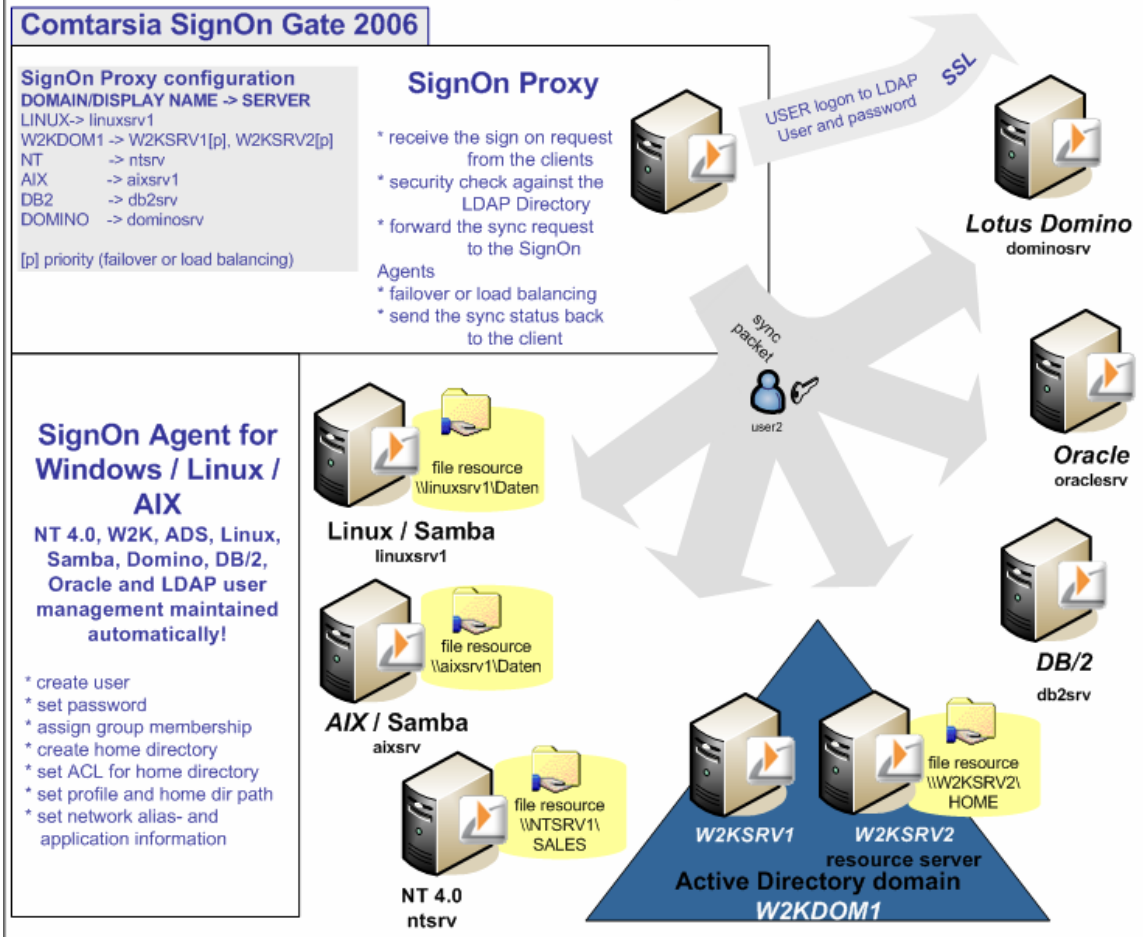
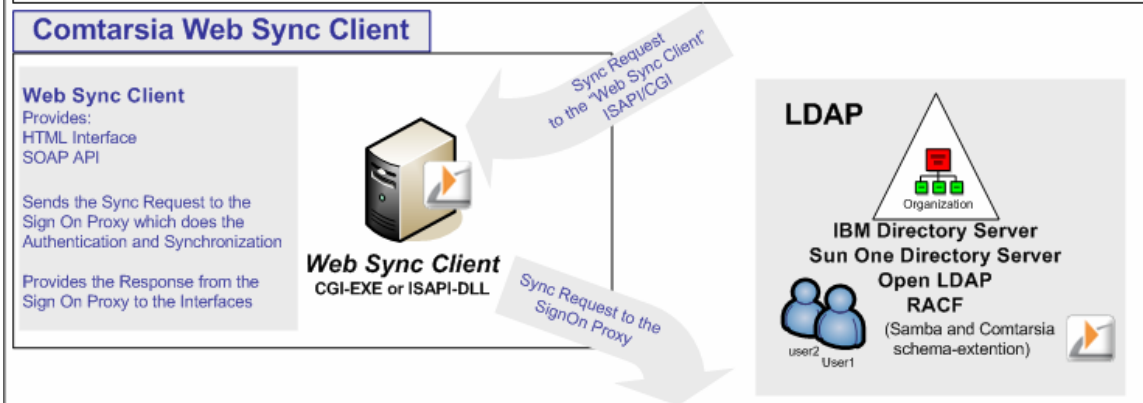
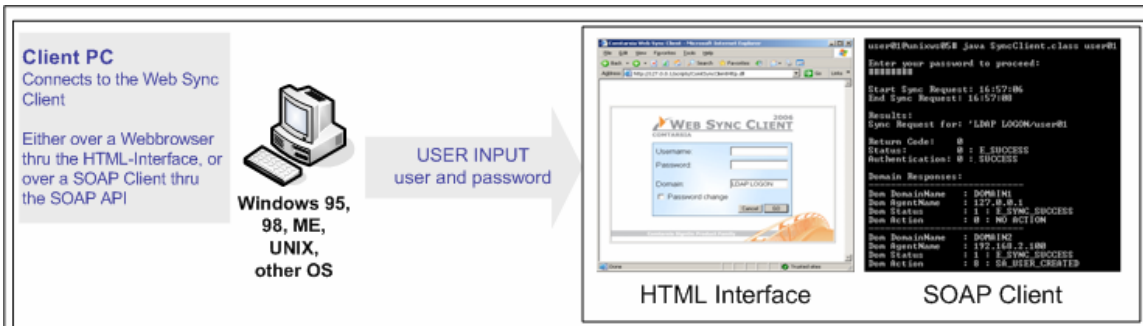
1.2 Funktionsbeschreibung

Der Comtarsia Web Sync Client stellt zwei verschiedene Betriebsmodi zur Verfügung:

- 1) ein Web Browser basierendes HTML Interface
- 2) ein SOAP-API.

Der „Web Sync Client“ leitet den „Sync Request“, welchen er entweder über das HTML Interface oder über das SOAP-API empfängt, weiter an den „Sign On Proxy“. Der „Sign On Proxy“ authentifiziert den Benutzer über den LDAP-Server und sendet bei erfolgreicher Authentifizierung „Sync Requests“ an alle „Sign On Agents“. Den Status sendet der „Sign On Proxy“ zurück an den „Web Sync Client“, welcher diesen dann entweder über das HTML Interface, oder über das SOAP-API zur ausgibt.





1.2.1 HTML Interface

Der Endbenutzer ruft über einen Web-Browser (Internet Explorer, Netscape, Konqueror, Mozilla Firefox) die durch den „Web Sync Client“ zur Verfügung gestellte HTML-Seite auf.

Dort hat der Benutzer die Möglichkeit mittels der Eingabe seiner Anmelde­daten (Benutzername und Passwort) die Synchronisation der Anmelde­daten durch den „Sign On Proxy“ zu veranlassen.

Ebenso erhält der Benutzer über das HTML-Interface die Möglichkeit sein Passwort (Eingabe: Benutzername, altes Passwort, neues Passwort) zu ändern.

Das HTML-Interface informiert den Benutzer über den/das Erfolg/Fehlschlagen des Anmelde- Synchronisations-Vorgangs und fordert diesen auf sein Passwort zu ändern, falls sich dieses in der „Password Expiration Time,“ befindet.

1.2.2 SOAP-API

Das SOAP-API stellt eine „Software Entwicklungs- Schnittstelle“ (API) zur Verfügung und bietet unseren Kunden dadurch die Möglichkeit, ein eigenes „Web Sync Client“ Interface zu entwickeln. Eine Vielzahl von Programmiersprachen unterstützen SOAP (Simple Object Access Protocol), wie z.B. Java, .NET, und somit ist sichergestellt, dass für eine Vielzahl von unterschiedlichen Betriebssystemen ein SOAP-Client programmiert werden kann.

SOAP Specifications

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

Apache SOAP

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

1.3 Installation

1.3.1 Generell

Um den „Comtarsia Web Sync Client“ verwenden zu können, benötigt man folgendes:

- Einen Webserver unter Windows (IIS5, IIS6)
- Einen Server mit „Comtarsia Sign On Proxy“
- Einen Server mit „Comtarsia Sign On Agent“
- Einen Directory Server (z.B. Netscape DS, IBM DS, OpenLDAP, ..)

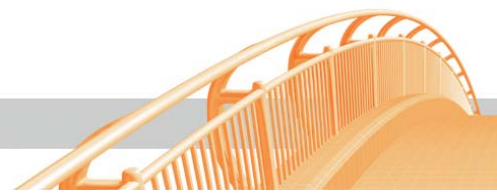
Natürlich kann man mehrere oder auch alle dieser Komponenten auf nur einen Server betreiben.

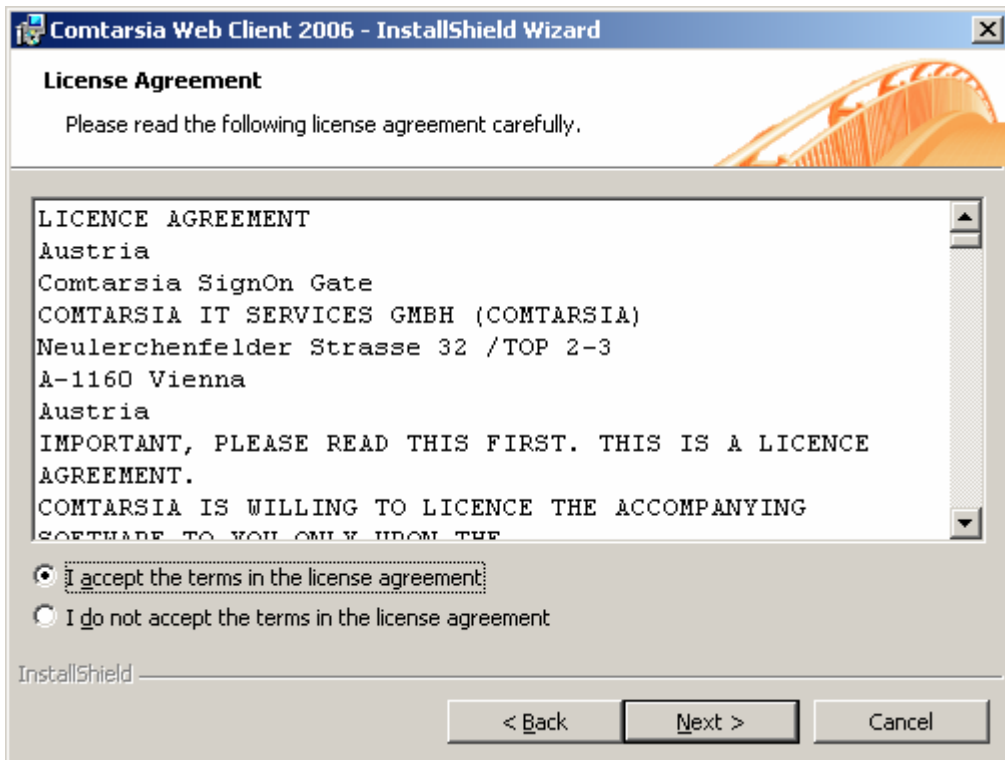
z.B.:

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

Stellen Sie sicher dass auf dem Server IIS installiert ist.

Starten Sie nun den Installer (WebClient_1.2.X.4.exe) um den WebClient zu installieren.





Geben Sie nun Ihren Namen und den Namen Ihrer Firma ein.

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

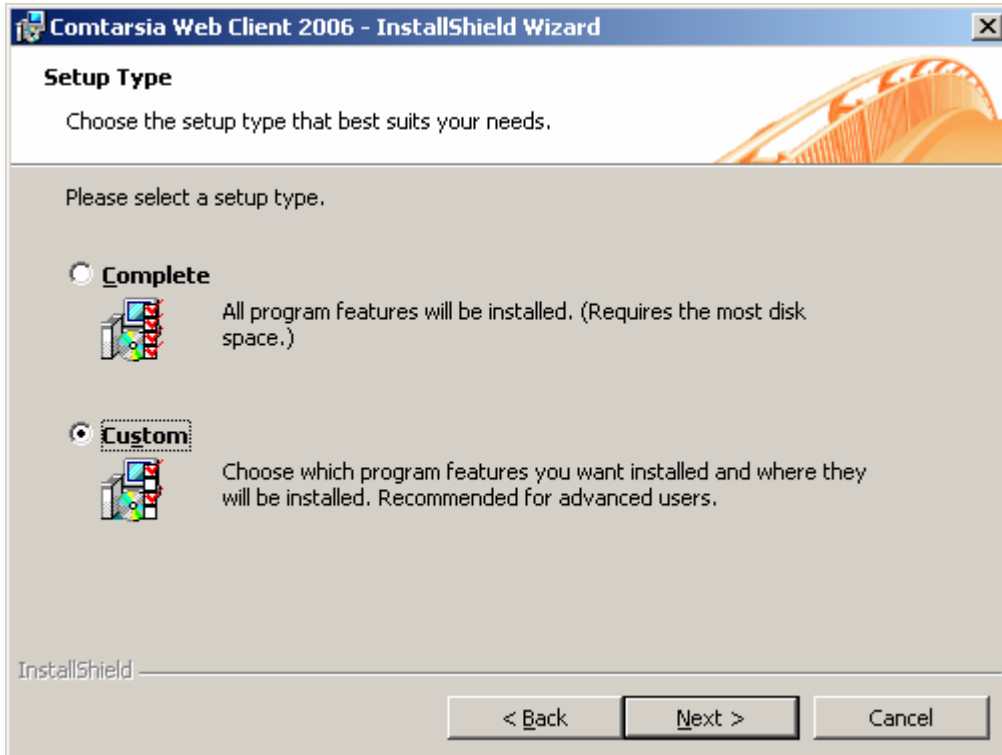
< Back Next > Cancel

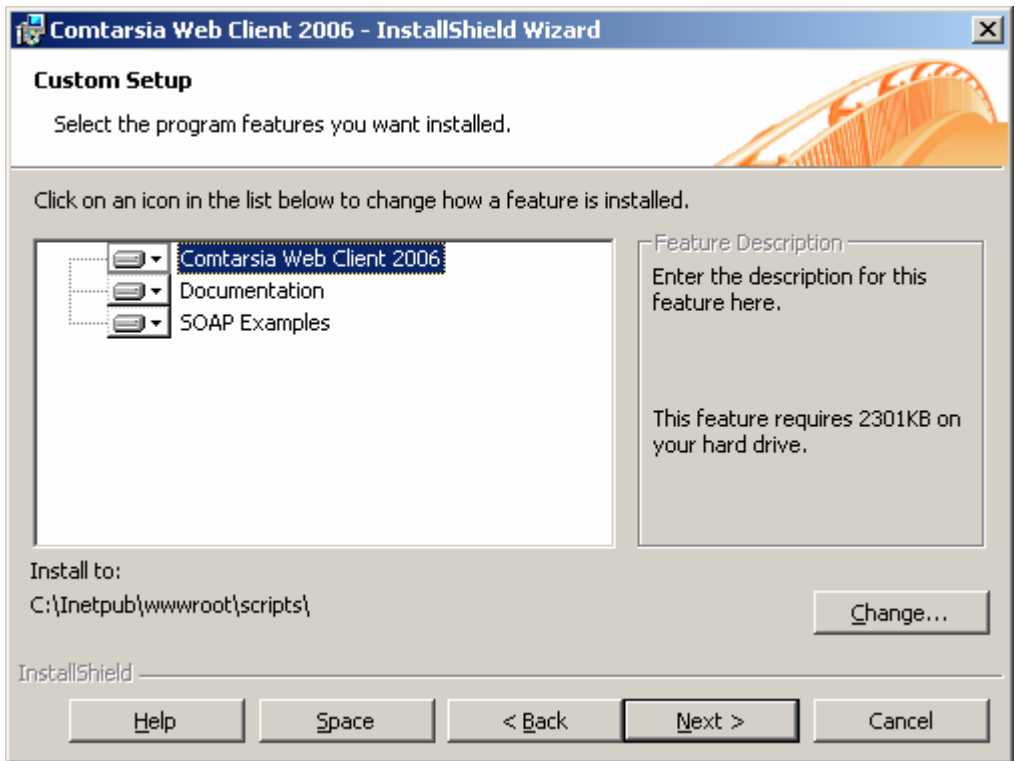
Klicken Sie nun auf „Next“.

Falls Sie IIS5 verwenden fahren Sie mit „**Installation unter IIS5**“ fort
Bei IIS6 fahren Sie mit „**Installation unter IIS6**“ fort.

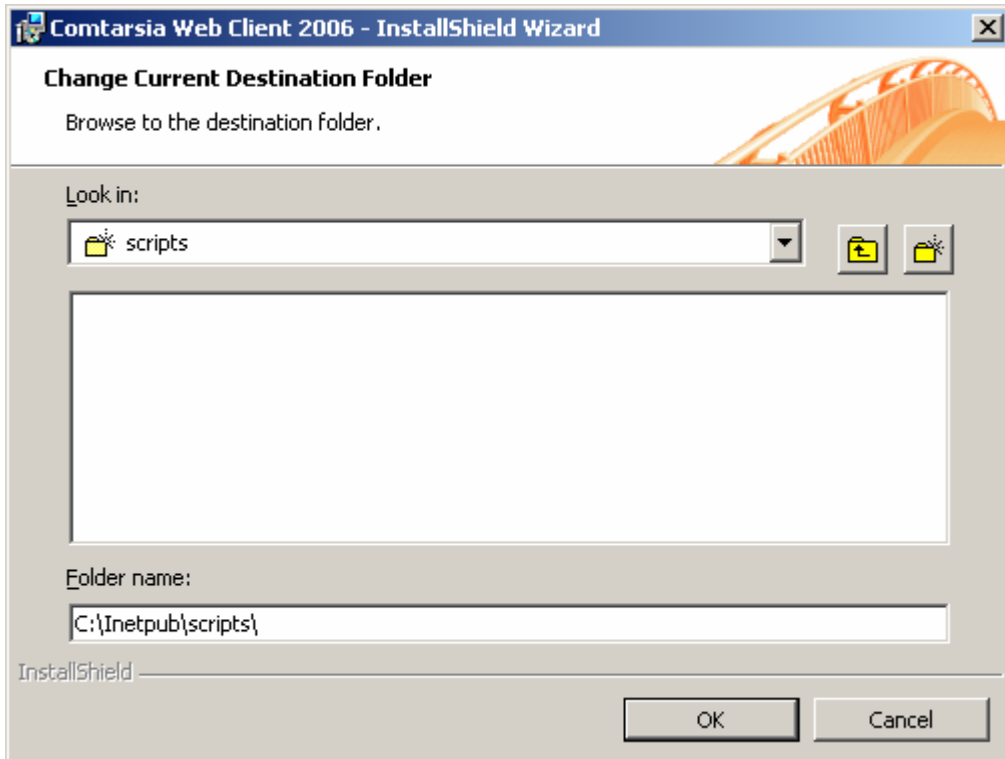
1.3.2 Installation unter IIS 5

Wählen Sie als „Setup Type“ Custom aus.

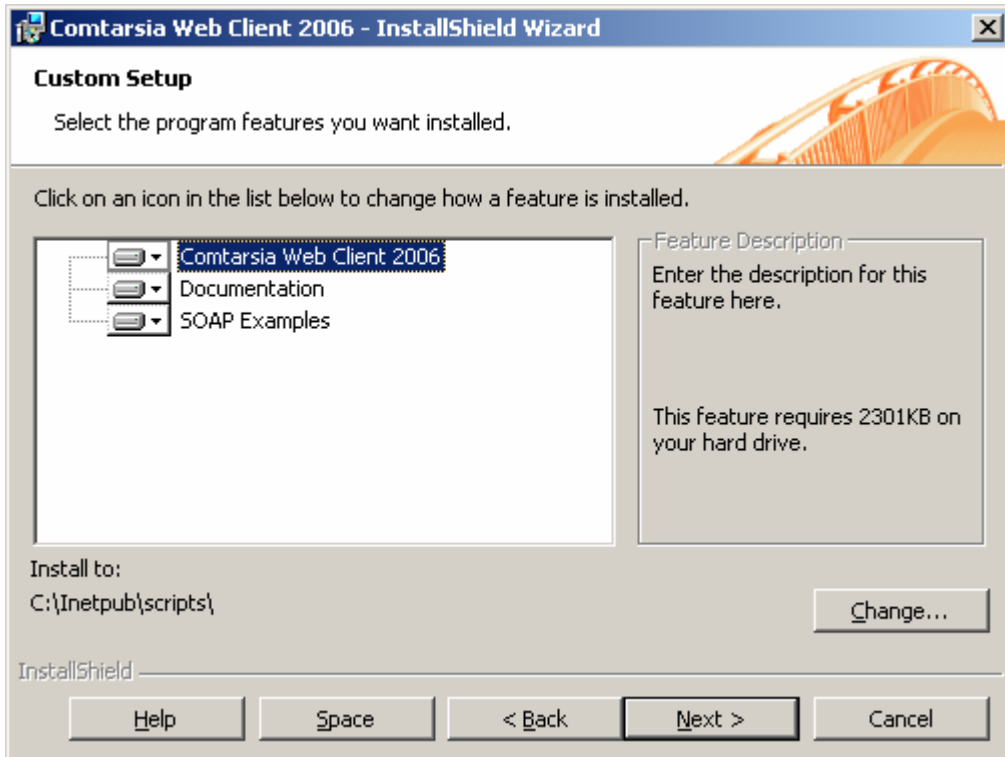




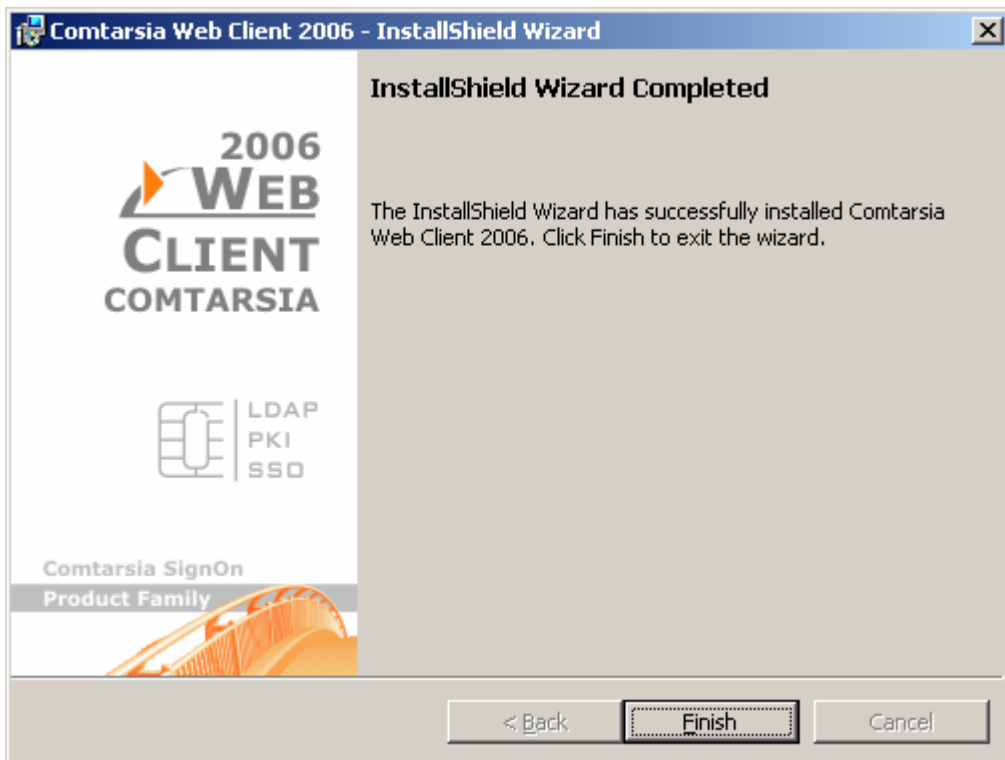
Ändern Sie den Pfad der „Comtarsia Web Client2006“ – Komponente auf:
„[IISDir]\Scripts“ z.B.: „c:\inetpub\scripts\“



Bestätigen Sie nun mittels „OK“.

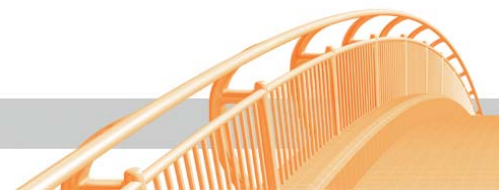


Klicken Sie nun auf „Next“ und anschliessend auf „Install“.



Die Installation der Binären Dateien, Dokumentation und SOAP-Beispiele ist nun abgeschlossen.

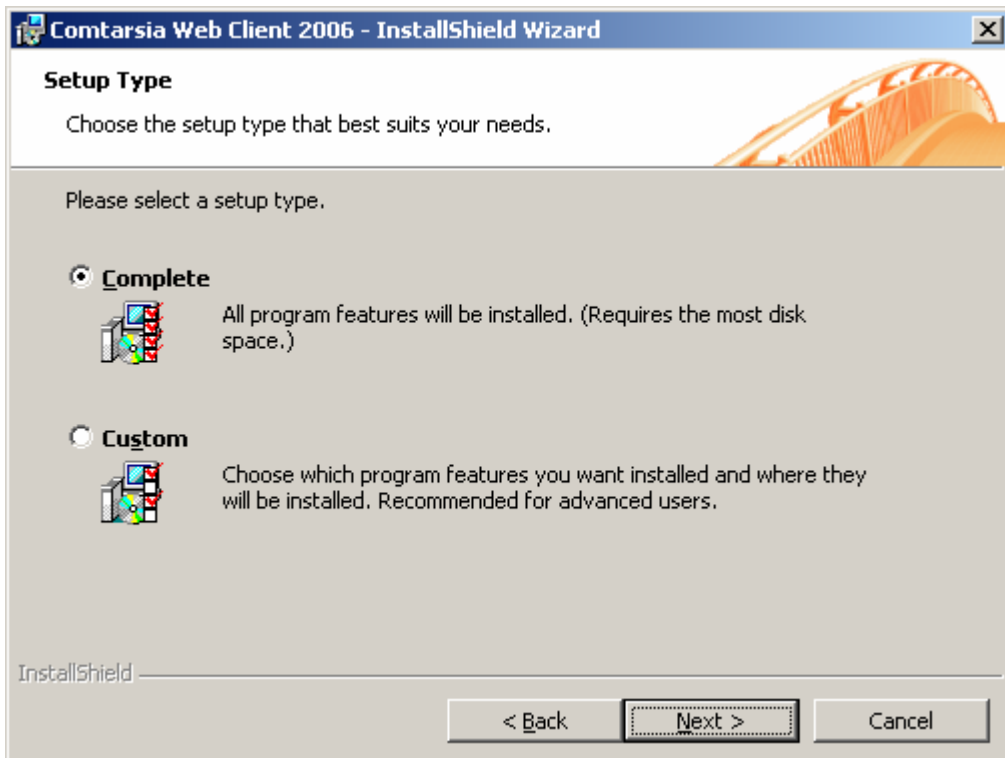
Für den IIS5 müssen keine weiteren Einstellungen vorgenommen werden. Fahren Sie nun mit dem Kapitel „**Konfiguration des WebClient**“ fort.



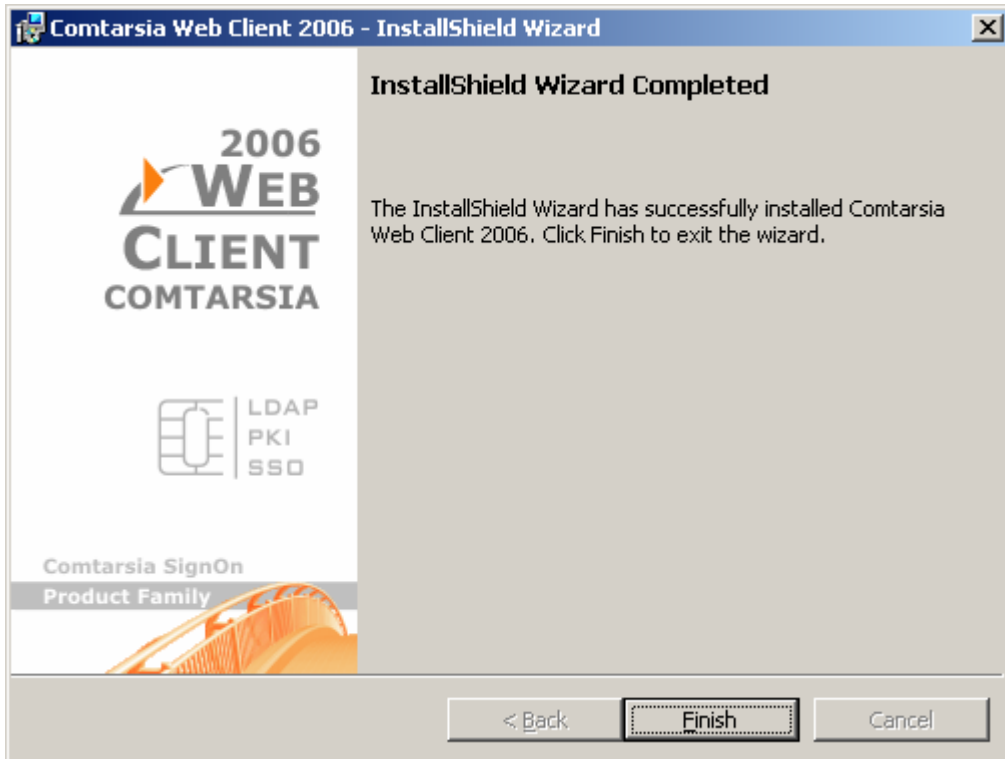
1.3.3 Installation unter IIS 6

Bei der Komplett-Installation werden der WebClient, die Dokumentation, Zertifikate (für die Kommunikation mit dem SignOn Proxy) und die SOAP-Programmbeispiele installiert.

Der WebClient wird in das IIS/Scripts – Verzeichnis installiert. Dokumentation, Programmbeispiele und Zertifikate werden unter „%programfiles%\Comtarsia\Web Client 2006\“ installiert.



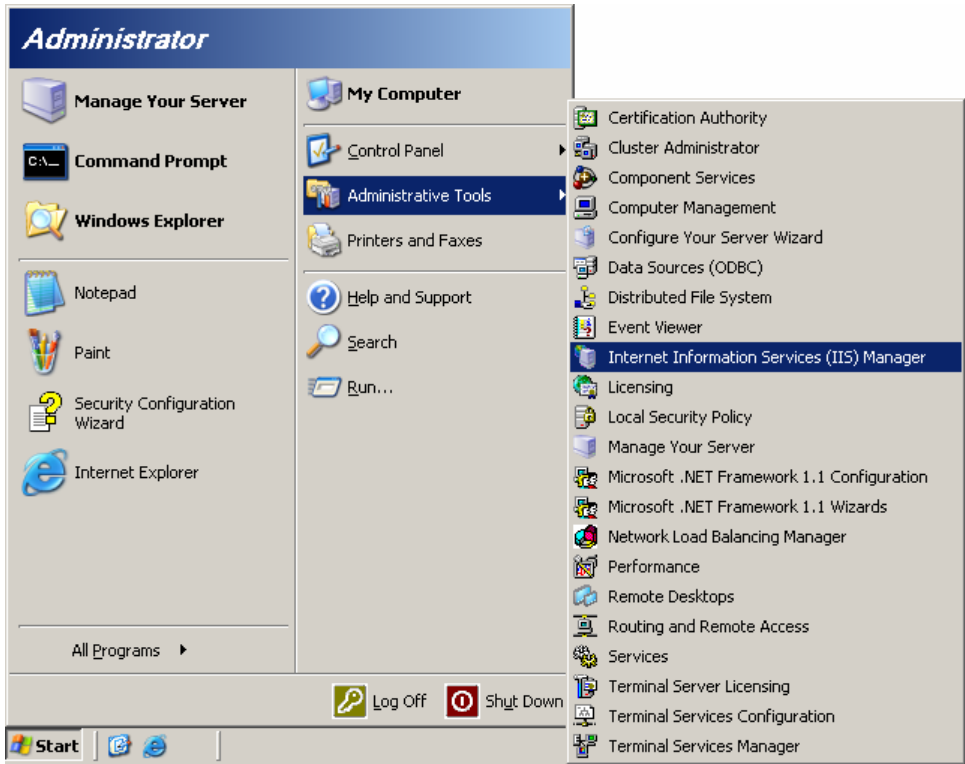
Klicken Sie nun auf „Next“ und anschliessend auf „Install“.

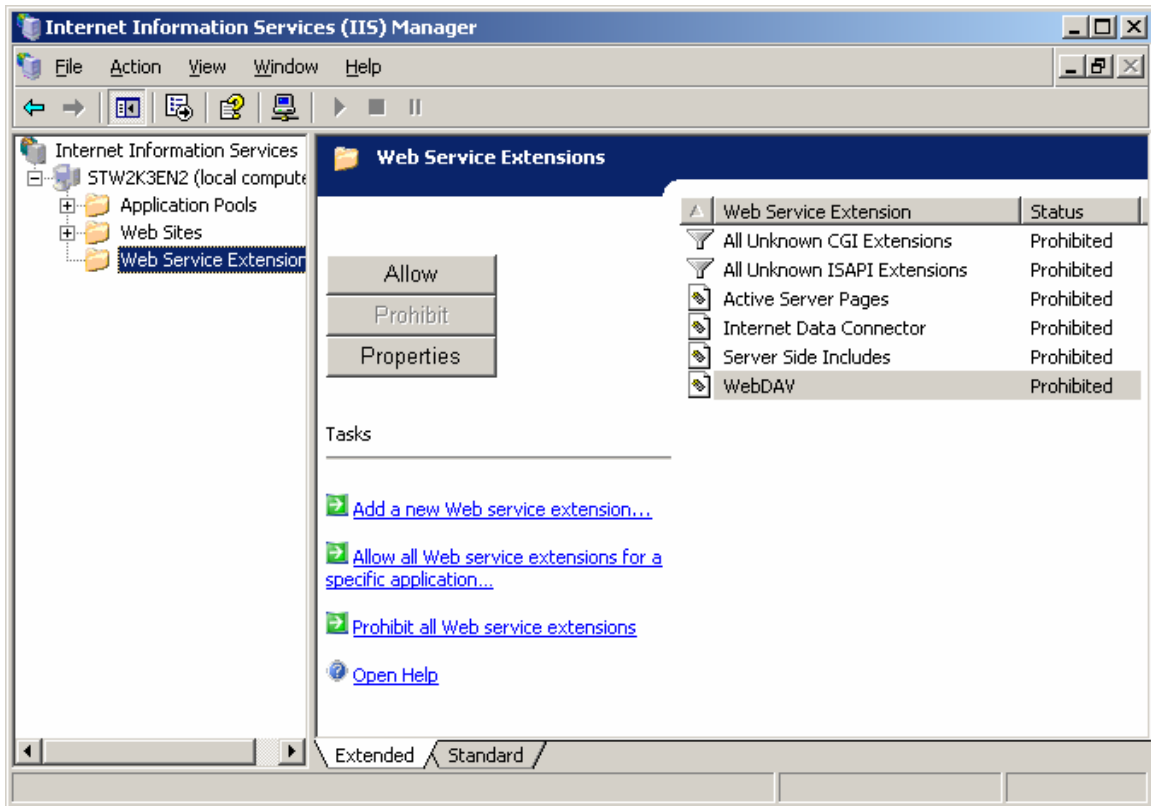


Die Installation der Binären Dateien, Dokumentation und SOAP-Beispiele ist nun abgeschlossen.

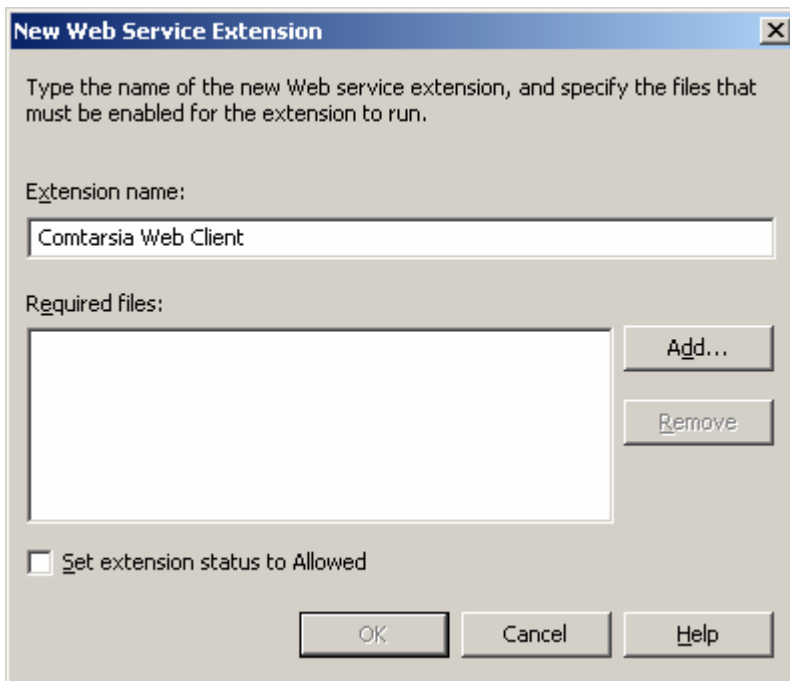
Nun muss noch eine Webservice Extension für den WebClient unter IIS6 erstellt werden.

Öffnen Sie hierfür die „IIS-Management Konsole“

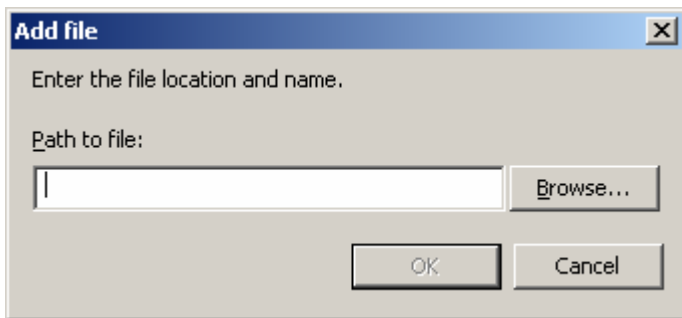




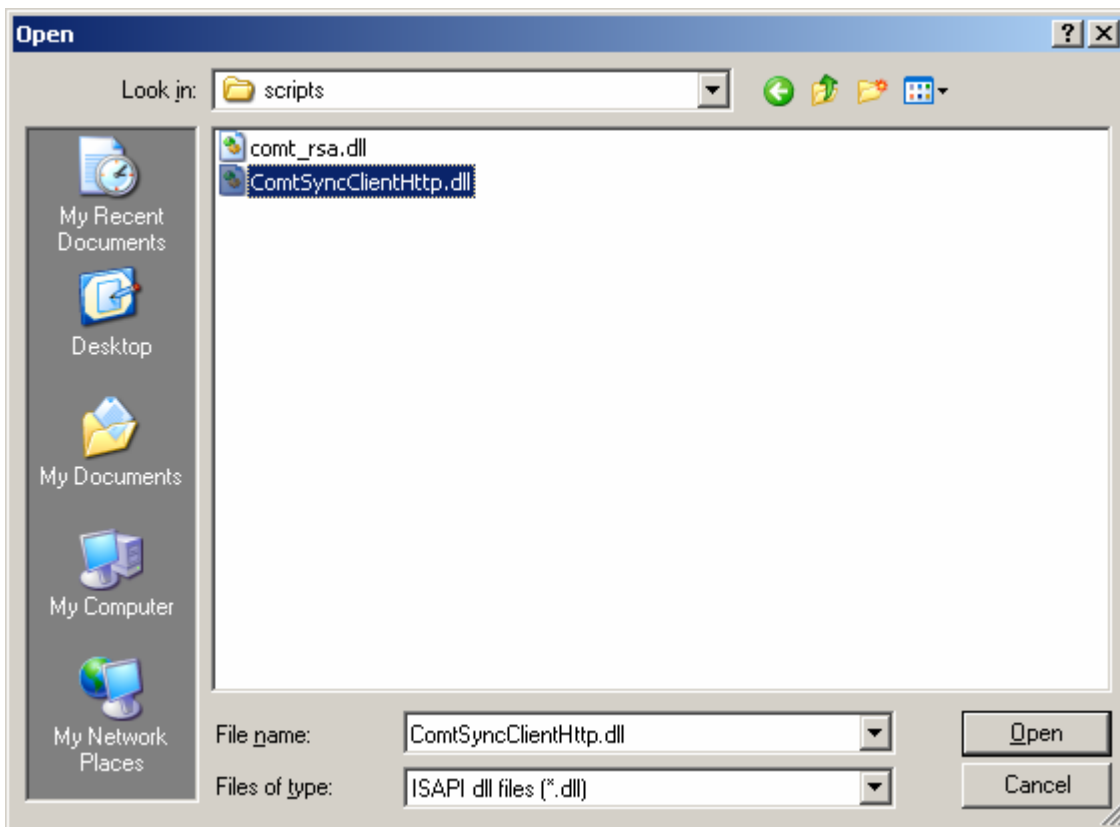
Klicken Sie nun auf „Web Service Extensions“ und anschliessend auf „Add a new Web service extension...“



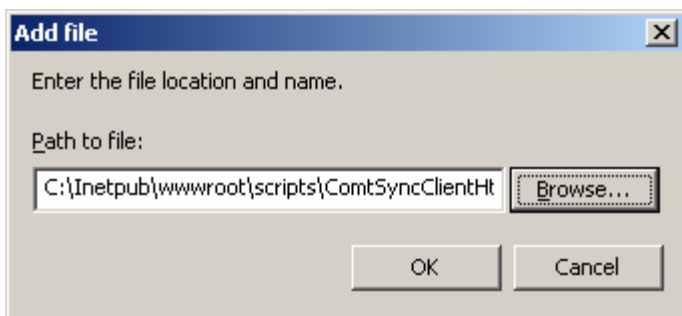
Geben Sie nun einen Namen für die Extension an, zB.: „Comtarsia Web Client“. Klicken Sie anschliessend auf „Add“.



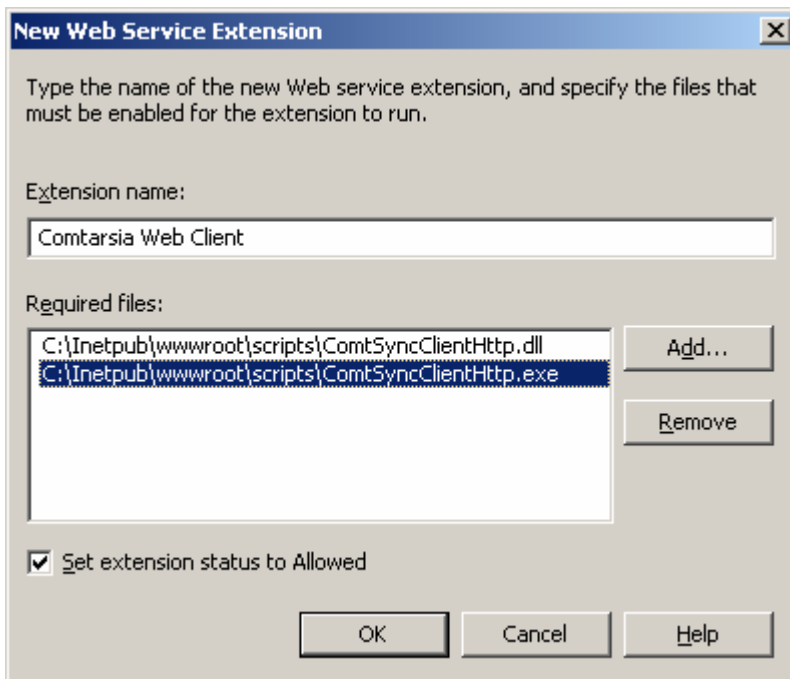
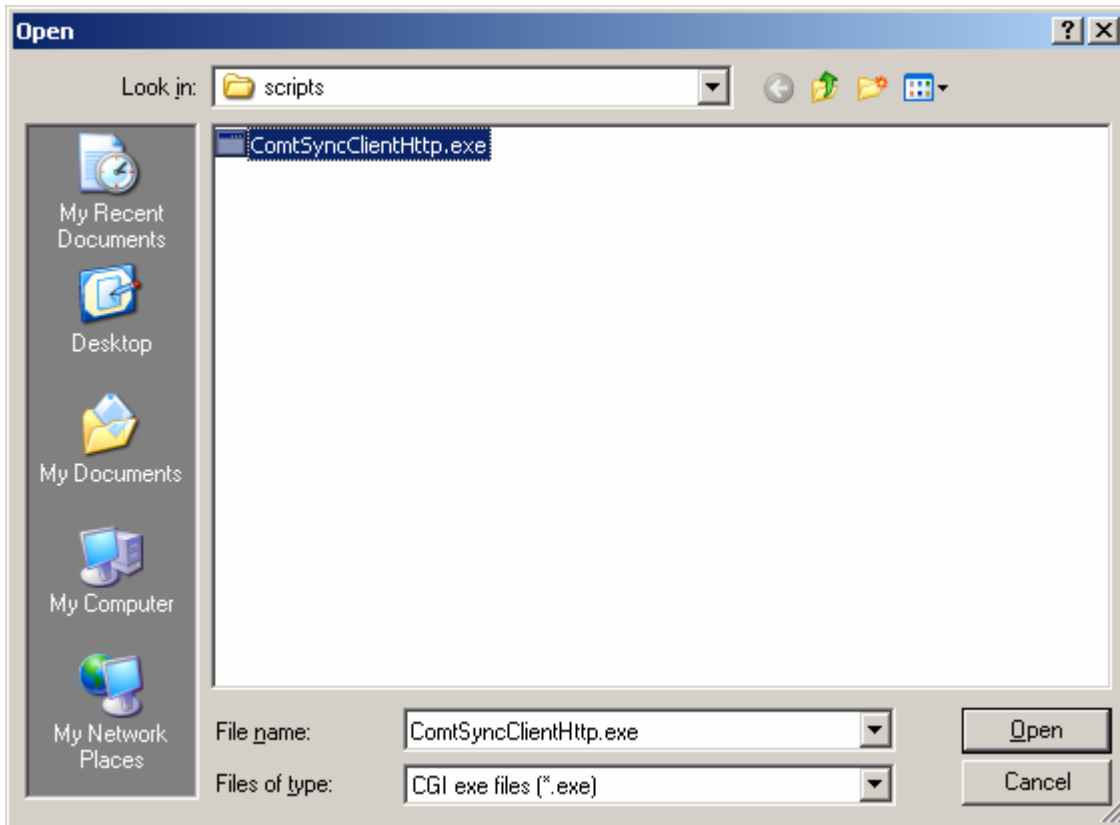
Klicken Sie auf „Browse“ und navigieren Sie zum Ordner der Comtarsia WebClient Binär-Dateien im IIS-Verzeichnis. zB.: „c:\inetpub\wwwroot\scripts\“



Wählen Sie nun die Datei „ComtSyncClientHttp.dll“ aus und klicken anschliessend auf „Open“, anschliessend auf „OK“.

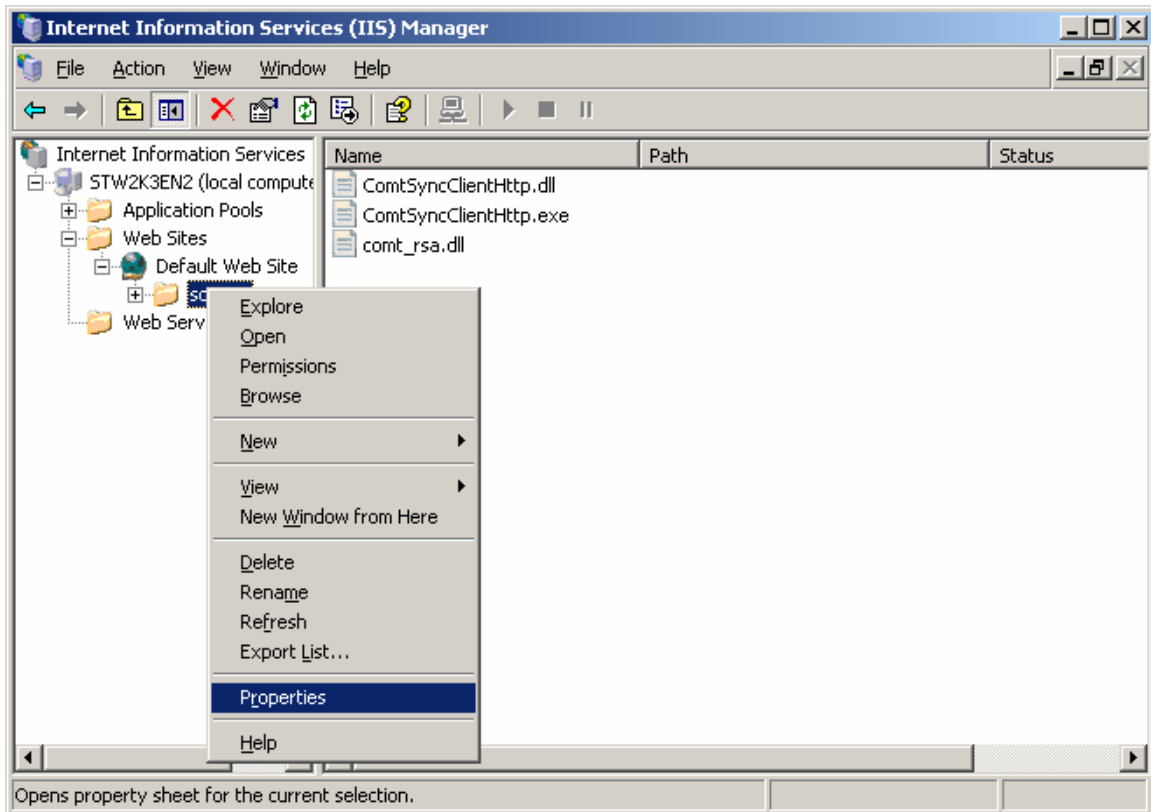


Wiederholen Sie nun diesen Vorgang für die Datei „ComtSyncClientHttp.exe“

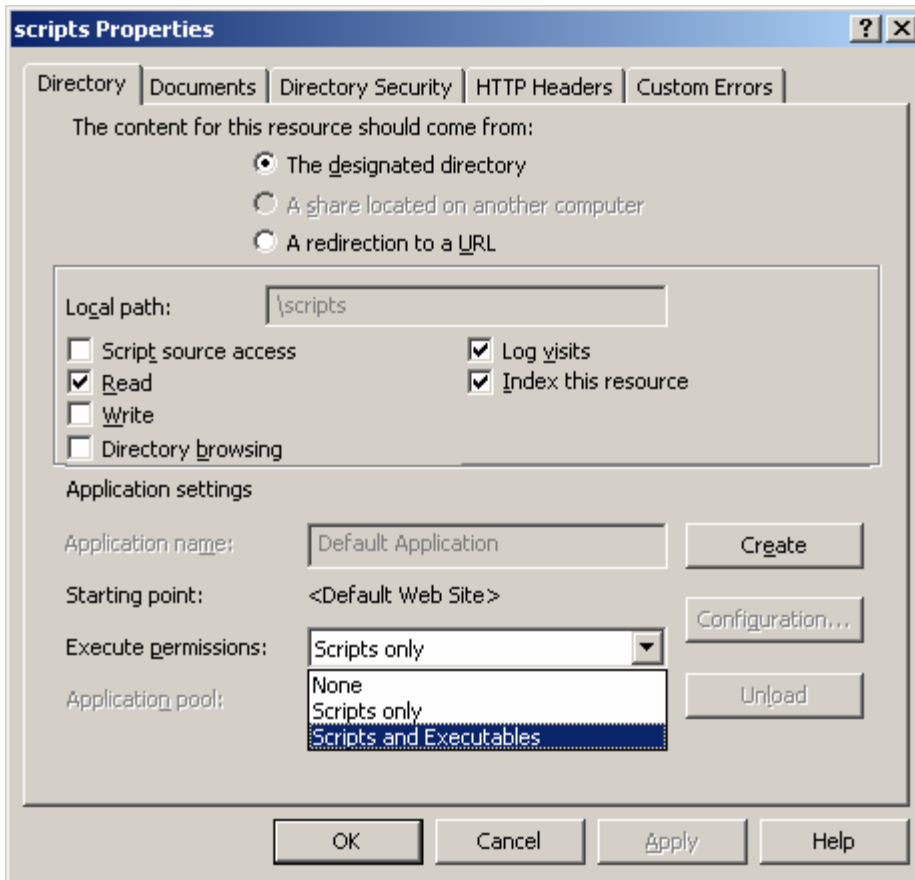


Aktivieren Sie nun die Checkbox „Set extension status to Allowed“ und bestätigen mit „OK“

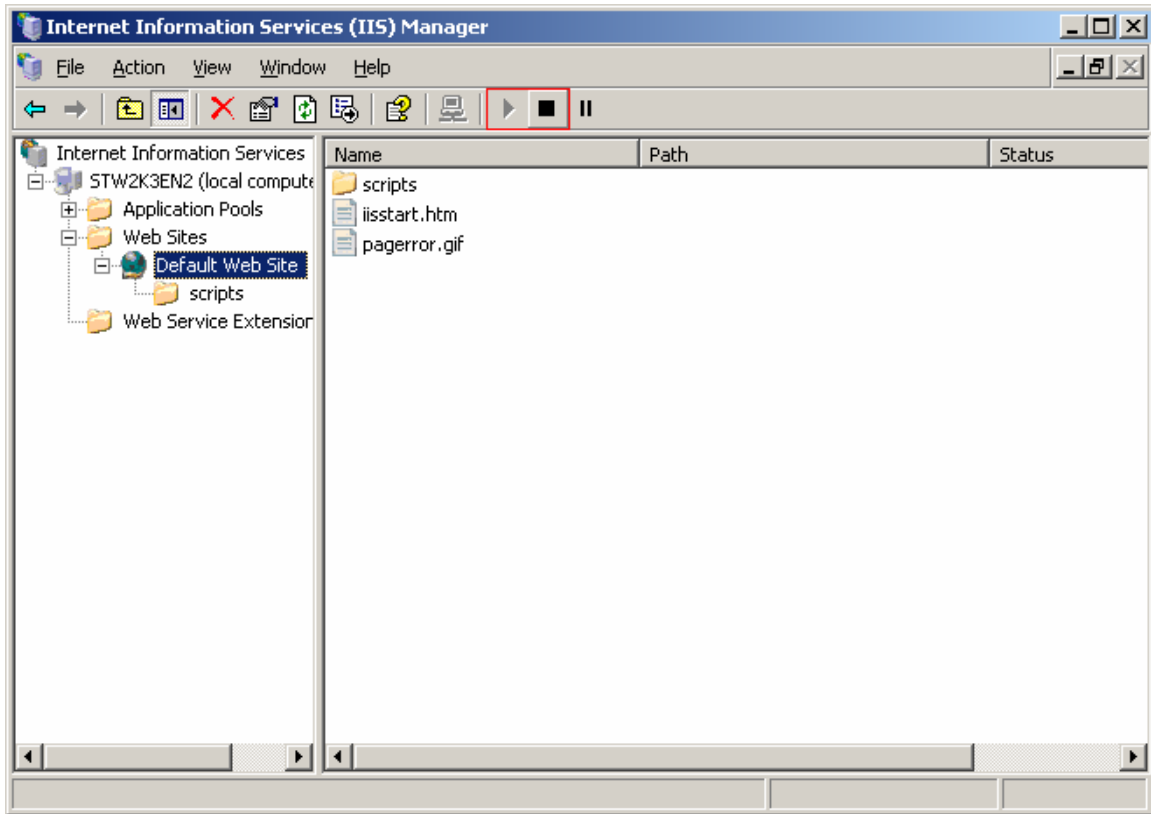
Navigieren Sie nun in der Managementkonsole zum Eintrag „Web Sites\Default Web Site\scripts“



Klicken Sie nun mit der rechten Maustaste auf „Scripts“ und anschliessend auf „Properties“.



Ändern Sie die „Execute permissions“ auf „Scripts and Executables“. Bestätigen Sie nun mittels „Apply“ und „OK“.



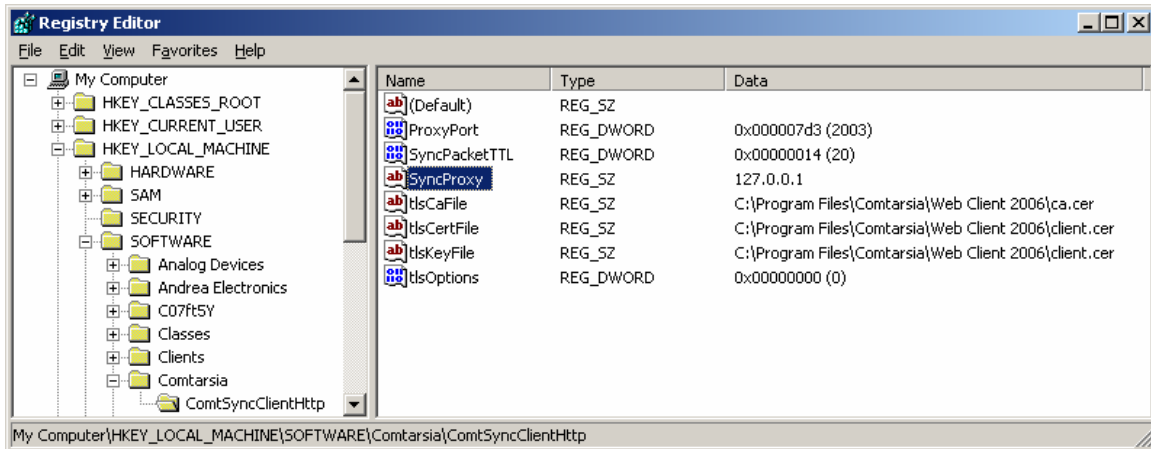
Starten Sie nun den IIS neu.

Nun ist der Installations-Part abgeschlossen, fahren Sie jetzt mit **„Konfiguration des WebClient“** fort.

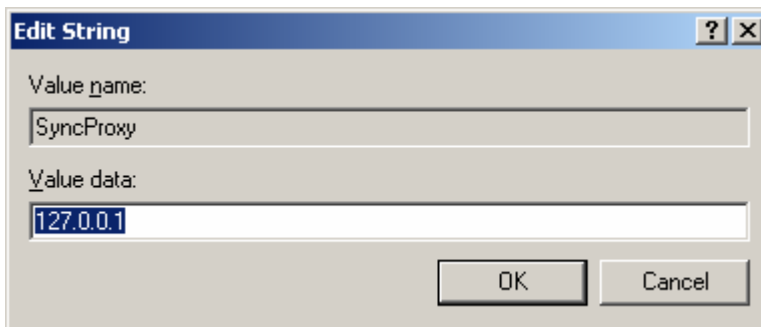
1.3.4 Konfiguration des WebClient

Falls der SignOn Proxy auf einen anderen Server als dem WebServer installiert ist, müssen Sie noch die IP-Adresse des Proxy-Servers anpassen.

Führen Sie hierfür Regedit aus.



Navigieren Sie nun zu dem Zweig:
„\HKEY_LOCAL_MACHINE\Software\Comtarsia\ComtSyncClientHttp“



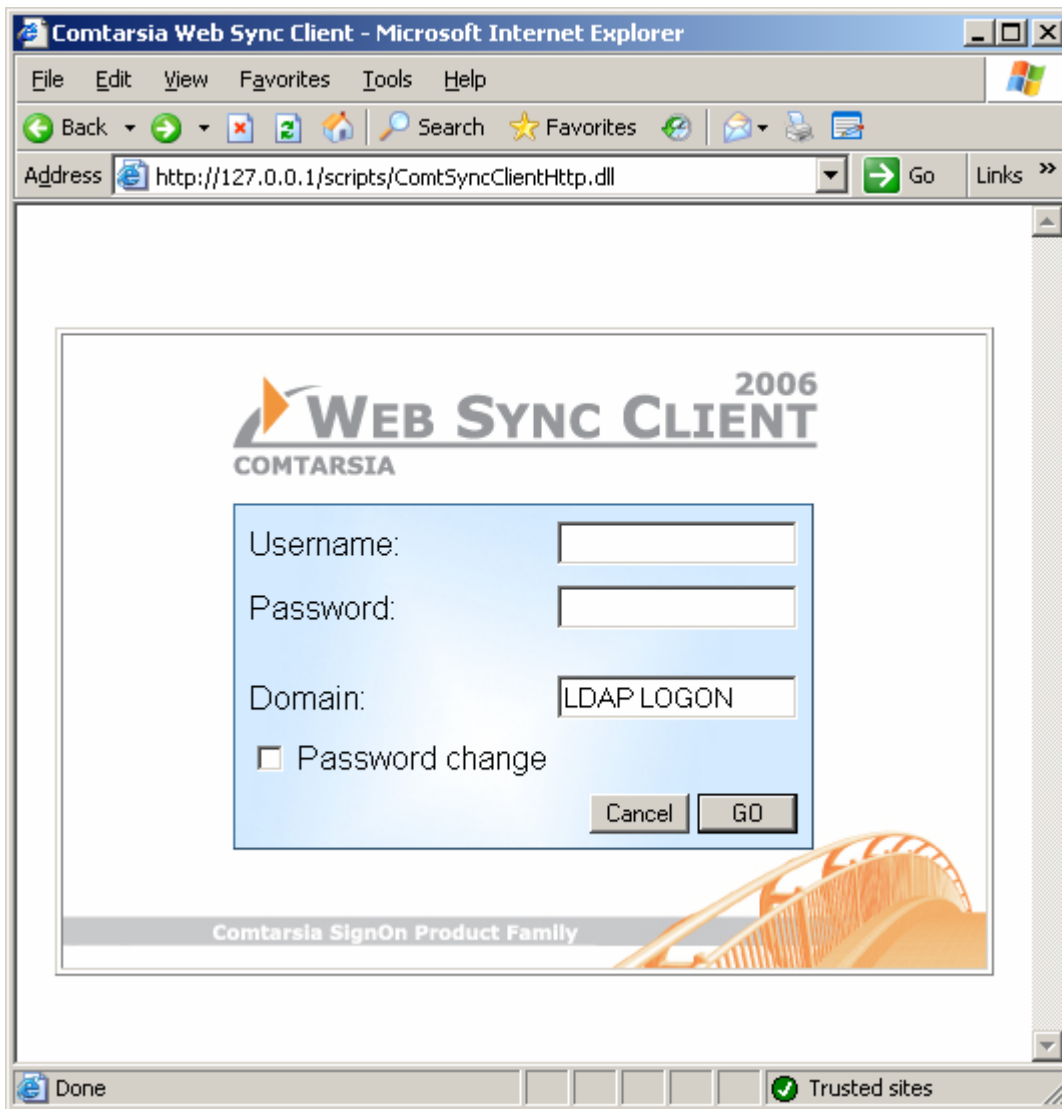
Ändern Sie den Wert „SyncProxy“ auf die IP Ihres Proxy-Servers.

Gratulation! Sie haben die Konfiguration des WebClients nun abgeschlossen!

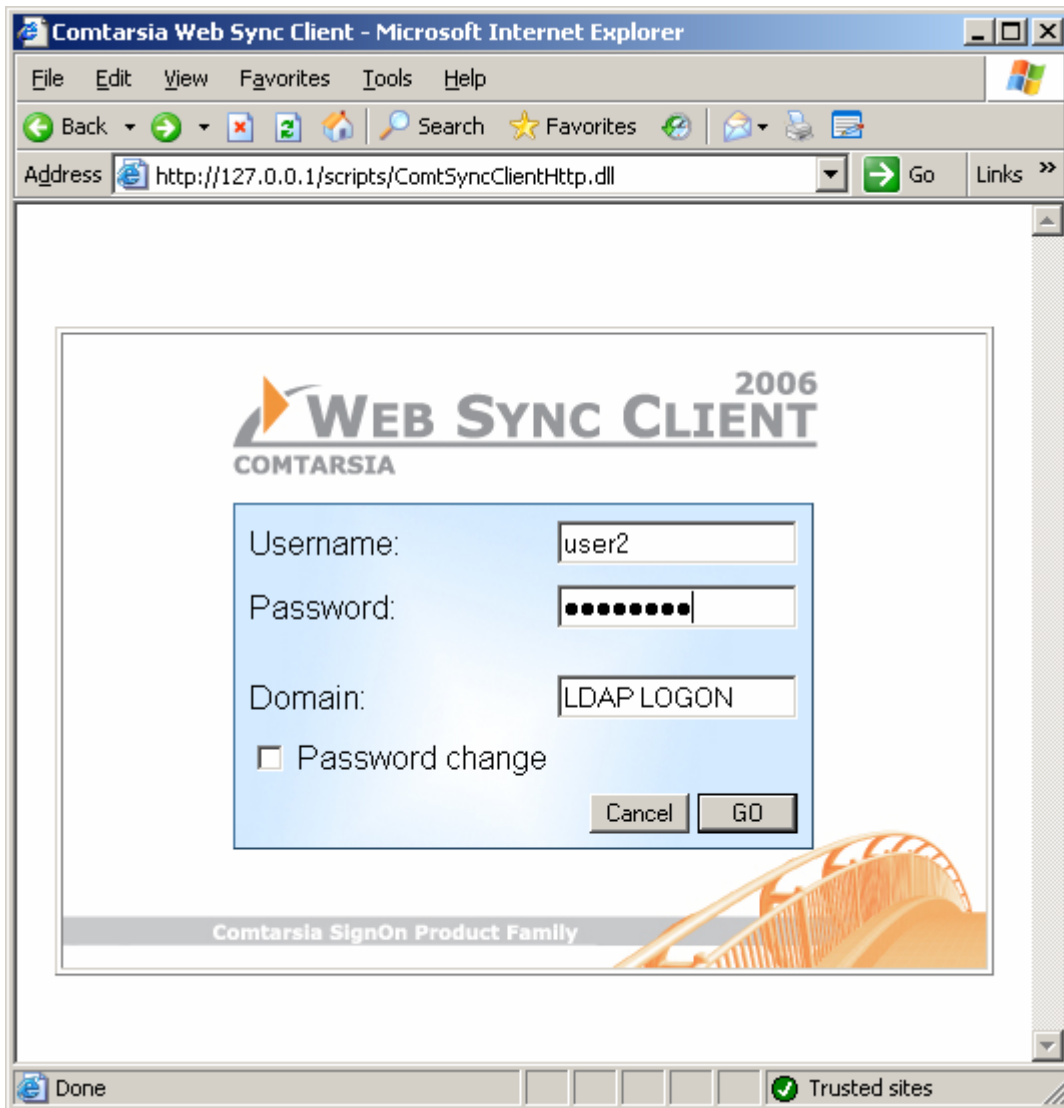
Um einen Benutzer über das „HTML Interface“ zu synchronisieren, gehen Sie zum Kapitel „Das HTML Interface“

1.3.5 Das HTML Interface

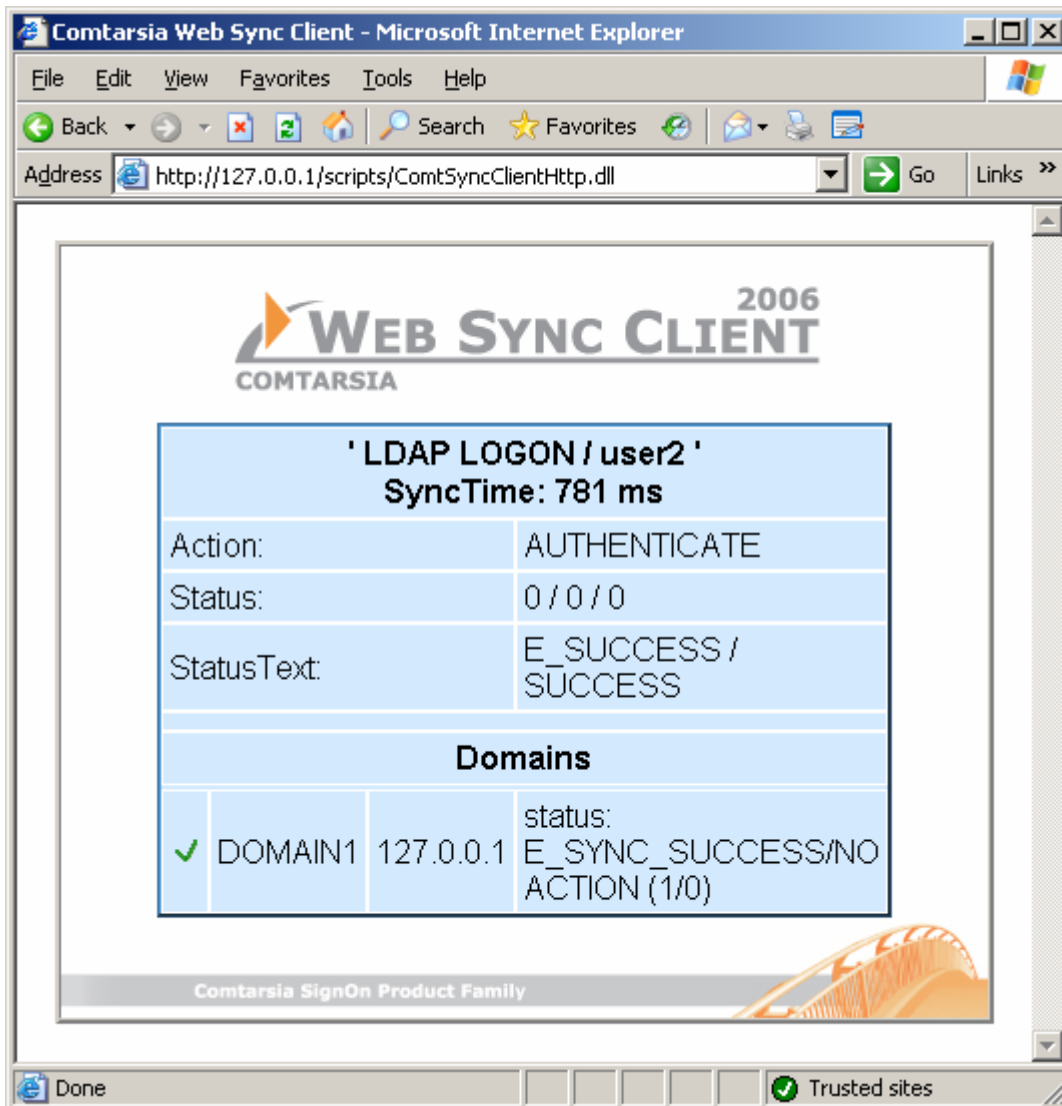
Zur Konfiguration des HTML Interface sind keine weiteren Schritte notwendig. Um das HTML Interface zu verwenden navigieren Sie einfach mit einem Web Browser zu der Adresse: `http://<nameIhresServers>/scripts/ComtSyncClientHttp.exe` oder
Adresse: `http://<nameIhresServers>/scripts/ComtSyncClientHttp.dll`



Um einen Benutzer zu synchronisieren, einfach Benutzernamen und Passwort des LDAP-Benutzerobjekts eingeben und mit „OK“ bestätigen.



[Eingabe von Benutzernamen und Passwort]



[Antwort über eine erfolgreiche Synchronisation]

1.3.6 SOAP API

Die Datei „ComtSyncClientHttp.exe“/ „ComtSyncClientHttp.dll“ fungiert gleichzeitig als SOAP-Server. Hierfür ist keine weitere Konfiguration notwendig.

WSDL:

http://<nameIhresServers>/scripts/ComtSyncClientHttp.[dll|exe]?WSDL

Für Beispiele zur Entwicklung eines SOAP Clients, lesen Sie das Kapitel „[SOAP Client Entwicklungs-Beispiele](#)“.

1.4 SOAP Client Entwicklungs-Beispiele

1.4.1 PHP SOAP-Client

Um einen PHP-SOAP-Client entwickeln zu können benötigt man:

Einen Webserver (zIIS6, Apache,...)

PHP5 www.php.net

Windows:

PHP 5.0.4 zip package

(beim Installer sind die Extensions nicht dabei, es wird aber die SOAP Extension benötigt)

Linux

PHP 5.0.4 Source

php mit SOAP-Extension erstellen.

Ändern sie die Konfiguration des Http-Webrowsers, sodass PHP-Scripts durch den PHP Interpreter verarbeitet werden. (siehe install.txt -->wird mit PHP mitgeliefert)

Fügen sie Folgende Zeile in die PHP.ini um die SOAP-Extension zu aktivieren:

unter Windows:

```
extension=php_soap.dll
```

unter Linux:

```
extension=php_soap.so
```

Beachten Sie ebenfalls, dass der Parameter "extension_dir" auf das Verzeichnis mit den extensions- gesetzt ist.

z.B.:

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

oder:

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

Starten Sie nun Ihren Webserver neu.

Kopieren Sie nun die von uns mitgelieferten "*.php" Dateien in ein Verzeichnis ihres Webrowsers.

z.B.:

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

Um zu überprüfen ob PHP, inklusive der SOAP Extension auf Ihren Webserver, funktionieren, navigieren Sie mittels eines Web-Browsers auf die (vorhin kopierte) phpinfo.php Datei.

z.B.:

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



Wenn alles funktioniert hat, sehen Sie nun eine PHP-Informationseite.
Weiter unten auf dieser Seite sind alle aktivierten PHP-Extensions aufgelistet.
Dort sollte nun auch die "SOAP"-Extension aufgelistet sein.

Editieren Sie nun die Dateien
phpsoap.php, phpsoap2.php
sodass die Variable \$wsdlUrl auf Ihre wsdl-Datei zeigt.
z.B.:

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

In der Datei phpsoap.php müssen Sie zusätzlich einen Benutzernamen und ein
Passwort
zum Testen eintragen.

Sie können nun mit dem Web-Browser zu einer dieser beiden Dateien navigieren um
die Anbindung zum "Web Sync Client" über SOAP mittels PHP zu testen.

Sie können die Dateien nun beliebig editieren und Ihren Bedürfnissen anpassen.

1.4.2 Java SOAP-Client

Um einen Java-SOAP Client erstellen zu können braucht man folgendes:

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) inkludiert in J2EE
<http://java.sun.com/products/javamail/downloads/index.html>

JavaBeans Activation Framework (activation.jar) inkludiert 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
(Alle ".jar" Dateien aus axis-1_2RC3\lib)

Installieren Sie zuerst das JDK.
Kopieren Sie anschließend die heruntergeladenen „.jar“ Dateien in das folgende
Verzeichnis:
[JDKInstallation]\jre\lib\ext
somit stehen die APIs zur Verfügung.

z.B.: c:\Program Files\java\jdk1.5.0_02\jre\lib\ext

Falls Sie ebenso das JRE (Java Runtime Environment) installieren, (wird mit dem jdk
mitgeliefert)
müssen Sie die ".jar" Dateien ebenso nach



[JREInstallation]\lib\ext
kopieren.

z.B.: c:\Program Files\java\jre1.5.0_02\lib\ext

Entpacken Sie nun die von uns mitgelieferte JavaSOAP.zip Datei in ein beliebiges Verzeichnis (z.B.: c:\devel\ComtSoap\)

Um in Java die Anbindung zum SOAP-Server vornehmen zu können öffnen Sie die Kommandozeile und wechselt in das Entwicklungsverzeichnis des SOAP Clients.

(z.B.: c:\devel\ComtSoap)

Erstellen Sie anschließend über die Kommandozeile mittels WSDL2Java die erforderlichen Klassen

java org.apache.axis.wsdl.WSDL2Java (WSDL-file-URL)

Der Parameter (WSDL-file-URL) muss auf die URL auf Ihrem Webserver zeigen.

z.B.:

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

Im Verzeichnis "com\comtarsia\signon\WebSyncClient" werden automatisch die notwendigen Klassen erzeugt.

Optional können Sie auch die Batch-Datei "createClasses.bat" verwenden.

(Die Datei muss sich im Entwicklungsverzeichnis befinden)

Dazu ändern Sie in der Datei die Zeile

set WSDLPath=http://server.comtarsia.com/ComtarsiaWebSyncClient_CGI.wsdl
auf den Pfad zu Ihrer wsdl Datei.

z.B.:

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

Um die Beispiele zu erzeugen, führen Sie einfach "buildSamples.bat" aus. Eventuell müssen Sie die Datei noch anpassen, sodass die "Java2SEPath" Variable auf Ihren Java2SE Installationspfad zeigt.

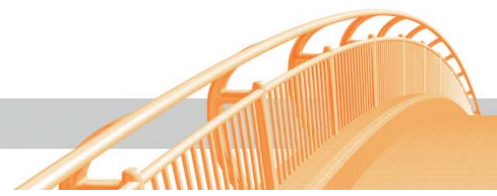
Anschließend können Sie die Beispiele mittels

```
java -cp "." comtSOAP Benutzername Passwort  
und
```

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

Das Beispiel comtSOAP ist so aufgebaut, dass Sie beim Aufrufen als Parameter Benutzername und Passwort übergeben.

Sie können den Quellcode der "ComtSOAP.java" und der "ComtSOAP2.java" an Ihre Bedürfnisse



anpassen.

Um die Dateien erneut zu kompilieren, führen Sie einfach wiederholt "buildSamples.bat" aus.

Um den SOAP-Client nur auszuführen (z.B.: auf einen Client-PC) benötigt man:
Sun:Java2RE

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

Anmerkung: Im Vergleich zum SDK (JDK) benötigt das JRE (Java Runtime Environment) weniger Festplattenspeicher

Und die ".jar" Dateien aus den folgenden Paketen.

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

Die heruntergeladenen ".jar" Dateien kopiert man in das folgende Verzeichnis:

[J2SEInstallation]\lib\ext

somit stehen die APIs zur Verfügung



Aufruf-Parameter:

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

// Values for 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

// Error Codes
#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 aquire context error

#define E_LASTERR_KEY (E_INTERNAL + 7)
// Some error with an RSA KEY occurred

#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

```

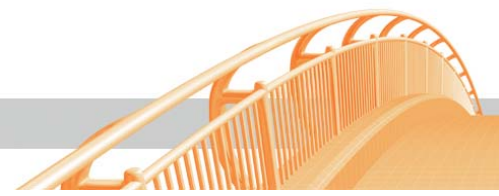


Return values for „sr.domains[i].status“:

```
#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“:

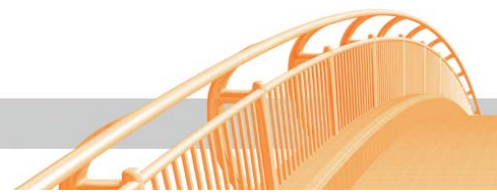
```
#define SA_USER_ENABLED        0x1
#define SA_USER_PW_SET         0x2
#define SA_USER_DELETED        0x4
#define SA_USER_CREATED        0x8
#define SA_USER_GRP_ADDED      0x10
#define SA_USER_GRP_DELETED    0x20
#define SA_USER_OU_MOVE        0x40
#define SA_USER_INFO_UPDATED   0x80
```



```

<?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>

```

