

net.data - Einrichtung des http-Servers für dynamische Inhalte

Aufsetzend auf der Konfiguration des http-Servers für statische Seiten richten wir nun den Server für „dynamische Inhalte“ ein. Darunter versteht man all jene Dinge, die nicht als fix und fertige Datei im IFS des Servers liegen, sondern „on demand“ zumindest teilweise erst erzeugt werden.

Das ist genau der Grund, warum man seine iSeries als http-Server einrichten sollte. Für einen http-Server, der nur statische Inhalte bietet, ist die Maschine sonst zu schade (oder „zu teuer“).

Diese Anforderung an einen http-Server ist nicht neu, man hat sich für eine der Lösungen den Namen „Common Gateway Interface“ einfallen lassen, als „cgi“ ist Ihnen das sicher schon in der Adresszeile Ihres Browsers begegnet.

In dieser Artikelserie lernen Sie den Einsatz eines speziellen CGI-Programms, nämlich „Net.Data“, kennen. Es handelt sich hier um eine leistungsfähige „Makrosprache“, die auf dem Server, also der iSeries ausgeführt wird. Der Quellcode wird interpretiert, das bedeutet, es muss (und kann) nichts umgewandelt werden. Die Performance ist trotzdem überraschend gut; mit großer Wahrscheinlichkeit werden Sie das auch auf Ihrem System so empfinden.

Nun geht es ans Einrichten; IBM-sei-Dank eine einmalige Arbeit.

Arbeiten in der 5250-Umgebung

Falls Ihre TechKnow-Instanz aus dem ersten Teil laufen sollte, drehen Sie sie bitte ab:

ENDTCPSVR SERVER(*HTTP) HTTPSVR(TECHKNOW)

Der erste Punkt ist die Erstellung einer CGI-Bibliothek, also einer Bibliothek, in die Sie Ihre CGI-Programm(e) stellen. Die Programme in dieser Bibliothek können aus dem Inter- bzw. Intranet aufgerufen werden.

CRTLIB LIB(TECHKNOW)TEXT('CGI-Bibliothek für http-Instanz TECHKNOW')

Gewähren Sie dem Benutzerprofil des http-Servers, QTMHHTTP, und dem Benutzerprofil für den Aufruf von CGI-Programmen, QTMHHTTP1, die Berechtigung, die Bibliothek zu verwenden.

GRTOBJAUT OBJ(TECHKNOW)OBJTYPE(*LIB)USER(QTMHHTTP QTMHHTTP1)AUT(*USE)

Erstellen Sie eine Quellendatei; wir brauchen Sie für ein kleines CL-Programm.

CRTSRCPF FILE(TECHKNOW/QPGMSRC)

Das CL-Programm heißt NETDATA und besteht aus 3 Zeilen:

```
PGM
CALL          PGM(TECHKNOW/DB2WWW)
ENDPGM
```

Listing 1 – Programm NETDATA Net.data - Listing 1

Erstellen Sie das Programm.

CRTBNDCL PGM(TECHKNOW/NETDATA)SRCFILE(TECHKNOW/QPGMSRC) SRCMBR(NETDATA)

Kopieren Sie das Programm DB2WWW in Ihre CGI-Bibliothek:

CRTDUPOBJ OBJ(DB2WWW)FROMLIB(QHTTPSVR)OBJTYPE(*PGM)TOLIB(TECHKNOW)

Gestatten Sie den Aufruf durch QTMHHTTP1:

GRTOBJAUT OBJ(TECHKNOW/DB2WWW) OBJTYPE(*PGM) USER(QTMHHTTP1) AUT(*USE)

CRTRCPF FILE(TECHKNOW/INI)RCDLEN(127)MBR(DB2WWW)TEXT('Einstellungen für Net.Data')

Es ist unbedingt notwendig, dass die Teildatei der INI-Datei DB2WWW heißt!!!

In diese Teildatei erfassen Sie nun diese 5 Zeilen:

```
MACRO_PATH           /WWW/TECHKNOW/MACROS
INCLUDE_PATH         /WWW/TECHKNOW/INCLUDES
EXEC_PATH            /QSYS.LIB;/QSYS.LIB/TECHKNOW.LIB
DTW_SHOWS_SQL       YES
DTW_SQL_ISOLATION   DTW_SQL_NO_COMMIT
DTW_INTERPRET_HTML_INPUT NO
```

Listing 2 – Teildatei DB2WWW in Datei TECHKNOW/INI Net.data - Listing 2

Die ersten 3 Zeilen sind Pfade zu Ihren Makros, zu Ihren Includes (alias Copies) und Programmen. (Die vierte Zeile ermöglicht es Ihnen, sich SQL-Statements anzeigen zu lassen. Dazu in Teil 3 mehr.)

Die fünfte Zeile ermöglicht die Ausführung von SQL-Statements auch ohne COMMIT-Steuerung (und dafür notwendige Journalisierung). Die sechste Zeile schließt eine potenzielle Sicherheitslücke.

Erstellen Sie nun das Verzeichnis, in das Sie Ihre Net.Data Makros ablegen werden:

md '/www/TechKnow/Macros'

CHGAUT OBJ('/www/techknow/macros')USER(QTMHHTTP1)DTAAUT(*RX)OBJAUT(*NONE)

und ein Verzeichnis für Ihre „Copies“, Ihre häufig verwendeten Funktionen.

md '/www/TechKnow/Includes'

und gewähren Sie auch hier QTMHHTTP1 das Recht zum Lesen und Ausführen.

Nun muß der CGI-Zugriff noch in unserer http-Server-Instanz eingerichtet werden. Ich könnte Ihnen nun hier etliche Bildschirmfotos präsentieren (die nach dem nächsten PTF eventuell schon wieder anders aussehen können).

Statt dessen verwenden wir auch zum Ändern der http-Konfiguration die 5250-Oberfläche – was Ihnen, ganz nebenbei gesagt, auch sicher eine halbe Stunde Zeit erspart. Außerdem werden Sie in nahezu allen Büchern und auf allen Webseiten, die sich mit Apache beschäftigen, immer nur die Konfiguration „wie sie wirklich ist“ sehen und nicht die web-basierte Konfiguration, die ist einzigartig auf der iSeries.

Editieren wir also mit

EDTF STMF('/www/techknow/conf/httpd.conf')

Sie sehen die so genannten „Direktiven“, die Ihren powered by Apache Server steuern.

Fügen Sie an das Ende die folgenden Zeilen an (der Editor funktioniert fast (so gut) wie SEU, „I“ ist auch dort „Insert“, also „Einfügen“ usw.):

```

# Aufruf von Net.Data Makros
ScriptAliasMatch (.*).cgi/(.*)$ /QSYS.LIB/TECHKNOW.LIB/NETDATA.PGM$1.ndm/$2

# Direkte Zugriffe auf das Bibliotheksdateisystem verbieten
<Location /QSYS.LIB>
  order allow,deny
  deny from all
</Location>

# Programmaufrufe in der CGI-Bibliothek
<Directory /QSYS.LIB/TECHKNOW.LIB>
  Options +ExecCGI
  SetHandler cgi-script
  ExpiresActive On
  ExpiresDefault "access plus 0 seconds"
  Order Allow,Deny
  Allow From all
</Directory>

```

Listing 3 – Apache-Konfiguration httpd.conf Net.data - Listing 3

Speichern Sie mit F2 und verlassen Sie den Editor mit F3. (Sie haben übrigens gerade eine Unicode-Datei bearbeitet; versuchen Sie das besser nicht mit einem PC-Editor, die Datei wäre danach ziemlich sicher kaputt! Wenn Sie sie auf andere Systeme übertragen wollen, verwenden Sie die Befehle SAV und RST.)

Nun zur Erklärung der Direktiven:

Die Zeile

```
ScriptAliasMatch (.*).cgi/(.*)$ /QSYS.LIB/TECHKNOW.LIB/NETDATA.PGM$1.ndm/$2
```

sagt dem http-Server, dass, wenn er die Zeichenfolge „.cgi/“ im Request des Clients (also im Normalfall des Browsers) findet, er keine statische Datei liefern soll, sondern stattdessen das Programm NETDATA in der Bibliothek TECHKNOW aufrufen soll.

Die Zeichen vor dieser Zeichenfolge werden mit (.* zusammengefasst und auf der rechten Seite des Ausdrucks im „in Wirklichkeit zu verwendenden“ request wieder eingesetzt. Das passiert mit den Zeichen dahinter ebenfalls, die werden dann mit \$2 eingesetzt.

Weiters findet hier quasi eine „Datei-Erweiterungs-Änderung“ statt. Während der Client ein „.cgi“ anfordert, wird auf ein Makro mit der Endung „.ndm“ zugegriffen. Aus der neutralen Endung kann ein Besucher, der Böses im Schilde führt, keine Schlüsse auf die verwendete Technologie zur Erstellung der Seiten am Server ziehen. (Wenn Sie wollen, können Sie sogar vortäuschen, Active Server Pages (.asp) oder Java Server Pages (.jsp) zu erzeugen.)

Das ist nur eine von vielen Möglichkeiten, die Aufgabe, ein Programm per http-Request aufzurufen, zu lösen. Aufgrund der Kürze und der Tarnungsmöglichkeit habe ich mich für diese entschieden.

```

<Location /QSYS.LIB>
  order allow,deny
  deny from all
</Location>

```

Diese XML-ähnliche Konstruktion (sie sieht nur ähnlich aus, es ist kein XML!) ist ein „Container“, man spricht auch von „Kontext“. Die Anweisungen zwischen den beiden Location-Tags gelten nur für Requests, die mit /QSYS.LIB beginnen. Hier wird verboten, auf Programme (oder andere Bibliotheksdateisystem-Objekte) per direktem Aufruf mit http://iseries:4000/qsys.lib/irgendeine.lib/irgendein.pgm zuzugreifen.

Damit wird gewährleistet, dass nur durch das ScriptAliasMatch auf Programme zugegriffen werden kann.

```
# Programmaufrufe in der CGI-Bibliothek
<Directory /QSYS.LIB/TECHKNOW.LIB>
  Options +ExecCGI
  SetHandler cgi-script
  ExpiresActive On
  ExpiresDefault "access plus 0 seconds"
  Order Allow,Deny
  Allow From all
</Directory>
```

Im nächsten Container, der sich auf das IFS-Verzeichnis /QSYS.LIB/TECHKNOW.LIB bezieht, wird der Zugriff erlaubt. Weiters wird jeder Request zu einem Programmaufruf.

Die Anweisung „ExpiresActive On“ weist Apache an, in seiner Antwort an den Client einen Verfallszeitpunkt einzubauen. „ExpiresDefault“ setzt diesen Verfallszeitpunkt auf den Zeitpunkt der Anforderung – die Seite verfällt sofort nach der Anforderung. Sinn des Ganzen ist es, zu verhindern, dass dynamische Seiten auf Proxies oder im Browser-Cache gehortet werden und eine veraltete Version angezeigt wird statt der aktuellen Version vom Server.

Das erste Makro

Am einfachsten ist es, wenn Sie eine Freigabe ihres IFS-Verzeichnisses mit einem Laufwerks-Buchstaben am PC verbinden. (Eine Freigabe können Sie mit dem iSeries Navigator erstellen.)

Erstellen Sie im Verzeichnis /www/TechKnow/Macros Ihr erstes Makro, nennen Sie es bitte „test.ndm“. Sie können das mit NotePad oder einem anderen Text-Editor tun; ich persönlich bevorzuge „UltraEdit“, aber das ist hauptsächlich eine Geschmackssache.

```
%HTML (START) {
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Net.Data Testseite</title>
</head>
<body>
Heute ist @DTW_rDATE(), es ist @DTW_rTIME().
<hr>
Ihre IP-Adresse ist @DTW_rGETENV("REMOTE_ADDR").
</body>
</html>
%}
```

Listing 4 – Makro test.ndm Net.data - Listing 4

Die Datei test.ndm ist größtenteils eine normale HTML-Datei.

HTML ist die Sprache des World-wide-web, nahezu alle Seiten, die Sie in einem Browser sehen, sind in dieser Sprache geschrieben.

Die erste und die letzte Zeile sind reine Net.Data-Anweisungen – sie kennzeichnen und begrenzen den Makro-Abschnitt „START“. (Sie können die Abschnitte taufen wie Sie wollen!) Die @DTW-Dinge sind Net.Data-Funktionen. Sie können auch selber Funktionen erstellen; die Net.Data-Funktions-Namen beginnen alle mit @DTW.

Alle anderen Dinge sind HTML-Anweisungen oder freier, anzuzeigender Text.

Eine HTML-Datei hat folgendes Gerüst:

`<!DOCTYPE>` gibt die verwendete HTML-Version an.

Mit `<html>` beginnt die Seite und mit `</html>` endet sie. Direkt unter `<html>` dürfen sich nur die beiden Tags `<head>` und `<body>` befinden. Im `<head>`-Bereich befinden sich Meta-Informationen über die Seite; hier nur der Seitentitel `<title>`, der normalerweise in der Titelleiste des Browserfensters angezeigt wird.

Als Einstieg und zur Vertiefung Ihrer HTML-Kenntnisse, aber auch als Nachschlagewerk ist das komplett deutschsprachige „SelfHTML“ von Stefan Münz sehr zu empfehlen. Es handelt sich hier um eine HTML-Dokumentation in – HTML. Sie können diesen Schatz unter <http://selfhtml.teamone.de> online abrufen, aber auch als Zip herunterladen, was bei häufiger Verwendung zu empfehlen ist.

Zwischen dem öffnenden `<body<-Tag` und dem schließenden `</body>-Tag` befindet sich der anzuzeigende Text. Die horizontale Linie `<hr>` ist einer der wenigen HTML-Tags, der keinen schließenden Partner hat, der häufigste ist `
`, die Zeilenschaltung.

Nach so viel Vorarbeit wollen wir nun sehen, ob die Sache auch funktioniert.

Starten Sie nun die TECHKNOW-Instanz wieder.

STRTCPVR SERVER(*HTTP)HTTPSVR(TECHKNOW)

Rufen Sie das Makro auf:

`http://!Series:4000/test.cgi/start`

Es sollte nun das Datum, die Uhrzeit und die IP-Adresse, von der aus Sie „surfen“, angezeigt werden:

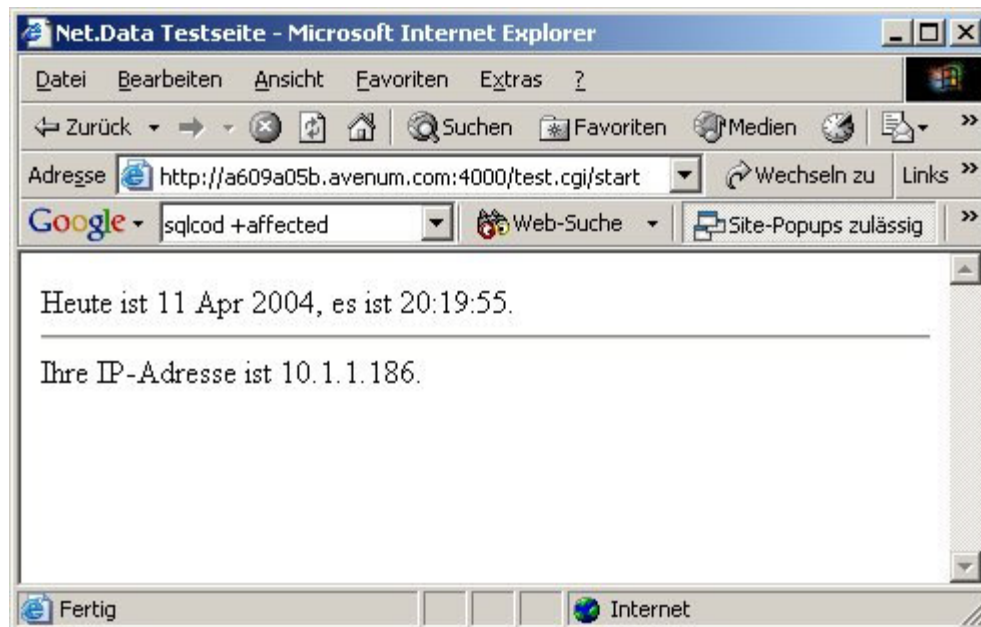


Abbildung 1 – Ausgabe von test.ndm

Sehen wir uns den Quelltext der Seite an (rechte Maustaste):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Net.Data Testseite</title>
</head>
<body>
Heute ist 11 Apr 2004, es ist 20:19:55.
<hr>
Ihre IP-Adresse ist 10.1.1.186.
</body>
</html>
```

Von den Net.Data-Funktionen ist nichts mehr zu entdecken, der Browser erhält ganz normales HTML.

Net.Data versteht auch rein gar nichts von HTML (oder XML oder .csv oder was auch immer Sie mit Net.Data erzeugen wollen. Net.Data liest seine Eingaben von „stdin“, wohin der http-Server Formulardaten schickt, geht das Makro durch und führt die Funktionen aus – den Rest lässt es unverändert und kümmert sich nicht darum und schickt es an „stdout“, von wo es der http-Server zurück an den Client schickt. Ob der Rest nun HTML ist oder ein anderes Dateiformat, das Sie über Ihren Webserver bereitstellen wollen – Net.Data kümmert das nicht. Es kümmert sich auch nicht darum, ob der Client ein Browser ist oder aber ein Programm (das kann auch ein RPG-Programm auf einer iSeries sein!). Da eröffnen sich viele Möglichkeiten, wenn Sie sich das mal durch den Kopf gehen lassen.

Wenn Sie die Anzeige aktualisieren (F5), werden Sie feststellen, dass sich die Uhrzeit ändert. Wie oft schaffen Sie es auf Ihrem PC pro Sekunde? Denn der PC und seine Anbindung sind hier die langsamste Komponente.

Wenn Sie Probleme haben sollten, prüfen Sie mal, ob der Server überhaupt läuft.

WRKACTJOB JOB(TECHKNOW)

Wenn der Server läuft, schauen Sie mal in die Joblogs der laufenden Jobs. Wenn Sie dort nicht fündig werden, suchen Sie nach der Ursache in den erzeugten Joblogs für Benutzer QTMHHTTP.

Wenn der Server zwar läuft, aber die Seite nicht angezeigt wird, prüfen Sie, ob die Berechtigungen für alle benötigten Objekte stimmen. QTMHHTTP1 muss die Bibliothek, die Programme, die INI-Datei und die IFS-Verzeichnisse (kompletter Pfad!) benutzen dürfen. (Das sollten Sie aber im Joblog sehen.)

Im nächsten Teil erfahren Sie, wie Sie mit SQL auf die Datenbank zugreifen und wie Sie „ganz normale“ RPG-Programme aufrufen können.

Den Autor Anton Gombkötö erreichen Sie unter
Avenum Technologie GmbH - Brigittenauer Lände 50-54/6, A-1200 Wien
Tel. (+43) 1/92101-148, e-Mail: ag@avenum.com