

Technische Anforderungen
zum Empfang
von XML-Nachrichten

25.11.2004

Peer Uwe Peters

W I S C O R E
BERATUNG ■ SERVICES ■ PRODUKTE

e-on

1 Inhaltsverzeichnis

1	INHALTSVERZEICHNIS	2
2	ZIEL DIESES DOKUMENTS	3
3	KONTEXT	3
4	SENDEWEG	4
5	ERREICHBARKEIT	4
6	SERVERKOMPONENTEN	5
7	BEISPIEL	5
7.1	PERL	5
7.2	APACHE WEBSERVER	5
8	BEISPIELPROGRAMM	6
8.1	PROGRAMMTEXT	6

2 Ziel dieses Dokuments

In diesem Dokument sind die technischen Voraussetzungen beschrieben, die notwendig sind um XML-Nachrichten zu empfangen. Die Verarbeitung dieser Nachrichten ist nicht Teil der Ausführungen. Hierzu bedarf es notwendigerweise der speziellen Kenntnis der versandten Nachrichten.

Im SAP EBP Umfeld werden verschiedenen Nachrichtenstrukturen benutzt, um Partnern beispielsweise Bestellungen und/oder Gutschriftnachrichten zuzustellen. Auch der „Rückweg“ ist zur Kommunikation durchaus denkbar (Lieferavis, Bestellbestätigung, etc.), wird aber in der Regel von den beteiligten Parteien nicht genutzt.

Ziel dieses Dokuments ist Kenntnis der technischen Verfahren und Zusammenhänge zu verschaffen, die notwendig sind XML-Nachrichten zu empfangen.

3 Kontext

In der Regel wird heute die in den Unternehmen vorhandene Verbindung zum Internet genutzt, um Nachrichten zu versenden. Im Falle von SAP EBP werden die Nachrichten abhängig vom genutzten Szenario über drei verschiedene Wege versendet:

1. Klassisches Szenario
2. Lokales Szenario, erweitert-klassisches Szenario
3. Szenario mit angebundener SAP XI - Komponente

Das klassische Szenario (1) unterscheidet sich vom lokalen (2) oder erweitert-klassischem (2) Szenario nur durch die verwendete Nachrichtenstruktur. In allen drei Fällen wird die Nachricht in einem SAP-System erzeugt und zu einem, an dem jeweiligen SAP System angeschlossenen, SAP Business Connector übermittelt. Der Business Connector (BC) empfängt die Nachricht und konvertiert diese in eine XML-Nachricht, um diese entweder direkt oder über einen Proxy-Server an den Empfänger zuzustellen. Auf die näheren technischen Details (wie die Nachricht an den BC übermittelt wird, wie der Empfänger ermittelt wird, etc.) wird an dieser Stelle nicht weiter eingegangen, da die Kenntnis dieser Details für das Ziel dieses Dokuments unerheblich ist. Ebenso wird auf die XI – Komponente (XI) nicht näher eingegangen. Hier nur so viel: Die XI ist eine Komponente über die die Erzeugung, Konvertierung, Verarbeitung (ausgehende wie eingehende Nachrichten) und Versendung von Nachrichten aus SAP-Systemen sehr facettenreich und variabel gesteuert werden kann.

Am Ende des Konvertierungsprozesses steht in den in diesem Dokument betrachteten Fällen jeweils eine XML-Nachricht die einem Partner zugestellt werden muss.

4 Sendeweg

Die zu versendenden XML-Nachrichten werden vom Sender (dies kann der BC, die XI oder aber auch ein beliebiger anderer Client, wie z.B. ein Web-Browser sein) per http-Upload verschickt. Die kann direkt geschehen oder aber über einen oder mehrere Proxy-Server. Diese Nachricht wird an einen erreichbaren Empfänger gesendet.

Technisch handelt es sich um einen http-Request bei dem die XML-Nachricht an den empfangenden Server übermittelt wird. Der Vorgang ist vergleichbar mit dem Senden von Daten aus z.B. einem Formular an einen Server. Der Empfang kann daher mittels einer einfachen CGI-Funktion realisiert werden.

5 Erreichbarkeit

Der empfangende Server muss im Internet erreichbar sein. Da der Sender aktiv wird und die Nachricht zu einem Zeitpunkt verschickt, den der Empfänger nicht kennt, muss der Empfangsserver ständig erreichbar sein.

Der Empfangsserver benötigt i.d.R. eine fest zugeordnete und aus dem öffentlichen Internet erreichbare (d.h. dort geroutete und keine RFC 1918) IP-Adresse.

Ist der Server nicht direkt (oder nicht direkt über eine Firewall) zu erreichen, können Techniken wie IP-Forwarding, NAT (Network Address Translation), SSH-Tunnel etc. zur Weiterleitung durch einen direkt erreichbaren Server problemlos eingesetzt werden. Eine solche Lösung ist immer dann einzusetzen, wenn dem Server selbst eine IP-Adresse für private Netze (RFC 1918 Adressen wie z.B. 192.168.0.1) zugeordnet ist.

Sofern der Zugriff auf dem Server über eine Firewall gesichert wird, ist an dieser für den gewählten und mit dem Sender abgestimmten Empfangsport ist mindestens das http-Protokoll freizuschalten.

6 Serverkomponenten

Üblicherweise wird der Empfang von Nachrichten über das Internet mit Hilfe eines Web-Servers realisiert. Auf diesem Webserver laufen dazu geeignete Anwendungen, die die XML-Nachricht entgegennehmen und diese speichern und/oder weiterverarbeiten.

Für die meisten Betriebssysteme gibt kostenlose Lösungen, um einen Webserver aufzusetzen. Weiterhin gibt es meistens ebenfalls kostenfreie Software, um die entsprechenden Anwendungen, die die Daten empfangen und weiterverarbeiten können zu betreiben und zu entwickeln.

Im folgenden werden wir anhand eines Beispiels mit kostenlos verfügbaren Komponente zeigen wie Nachrichten empfangen und gespeichert, bzw. per Mail weiterversendet werden können.

7 Beispiel

7.1 Perl

Perl ist eine Scriptsprache die häufig eingesetzt wird, um Anwendungen für einen Webserver zu entwickeln. Eine Perl-Installation ist unter <http://www.perl.com/download.csp> für Windows- und Unix-Systeme zu finden. Dort sind ebenfalls Verweise auf weitere Versionen für andere Betriebssysteme wie beispielsweise Macintosh finden.

7.2 Apache Webserver

Der Apache Webserver ist unter der Adresse <http://httpd.apache.org> zu finden. Hier existieren für verschiedenste Plattformen sehr ausgereifte, einfach zu installierende und einzusetzende Versionen des gerne genutzten und stabilen Webservers „httpd“.

8 Beispielprogramm

Ein Beispielprogramm, das zusammen mit dem unter Abschnitt 7.2 genannten Servers und der Sprache Perl funktioniert ist im folgenden Abschnitt exemplarisch dargestellt.

8.1 Programmtext

```

1  #!/usr/local/bin/perl
2  open (DATEI,">> ../logs/demo.log");
3  $datum = localtime(time);
4  read(STDIN, $Daten, $ENV{'CONTENT_LENGTH'});
5  print DATEI $datum;
6  print DATEI "\n";
7  print DATEI "Content length:";
8  print DATEI $ENV{'CONTENT_LENGTH'};
9  print DATEI "\n";
10 print DATEI $Daten;
11 print DATEI "\n";
12 close DATEI;
13
14 use Mail::Mailer;
15 $type = 'sendmail';
16 $mailprog = Mail::Mailer->new($type);
17 %header = ('To' => 'bbp@wiscore.de','From' => 'bbp@wiscore.de','Cc' => 'peer.uwe.peters@wiscore.de','Subject' =>
18 'Nachricht von Wiscore Receive @ Metro-Lieferant'
19 );
20 $mailprog->open(\%header);
21 print $mailprog "Wir haben Ihre Nachricht erhalten.\n";
22 print $mailprog $datum . "\n";
23 print $mailprog "Email: " . $ENV{'HTTP_FROM'} . "\n";
24 print $mailprog "Remote IP: " . $ENV{'REMOTE_ADDR'} . "\n";
25 print $mailprog "Remote Host: " . $ENV{'REMOTE_HOST'} . "\n";
26 print $mailprog "Remote User: " . $ENV{'REMOTE_IDENT'} . "\n";
27 print $mailprog "Remote User Name: " . $ENV{'REMOTE_USER'} . "\n";
28 print $mailprog "Daten:" . "\n";
29 print $mailprog $Daten;
30 $mailprog->close;
31
32 print "Content-type: text/html\n\n";
33 print "<html><head><title>CGI-Feedback</title></head>\n\n";
34 print "<body>Alles OK!";
35 print "</body></html>\n\n";

```