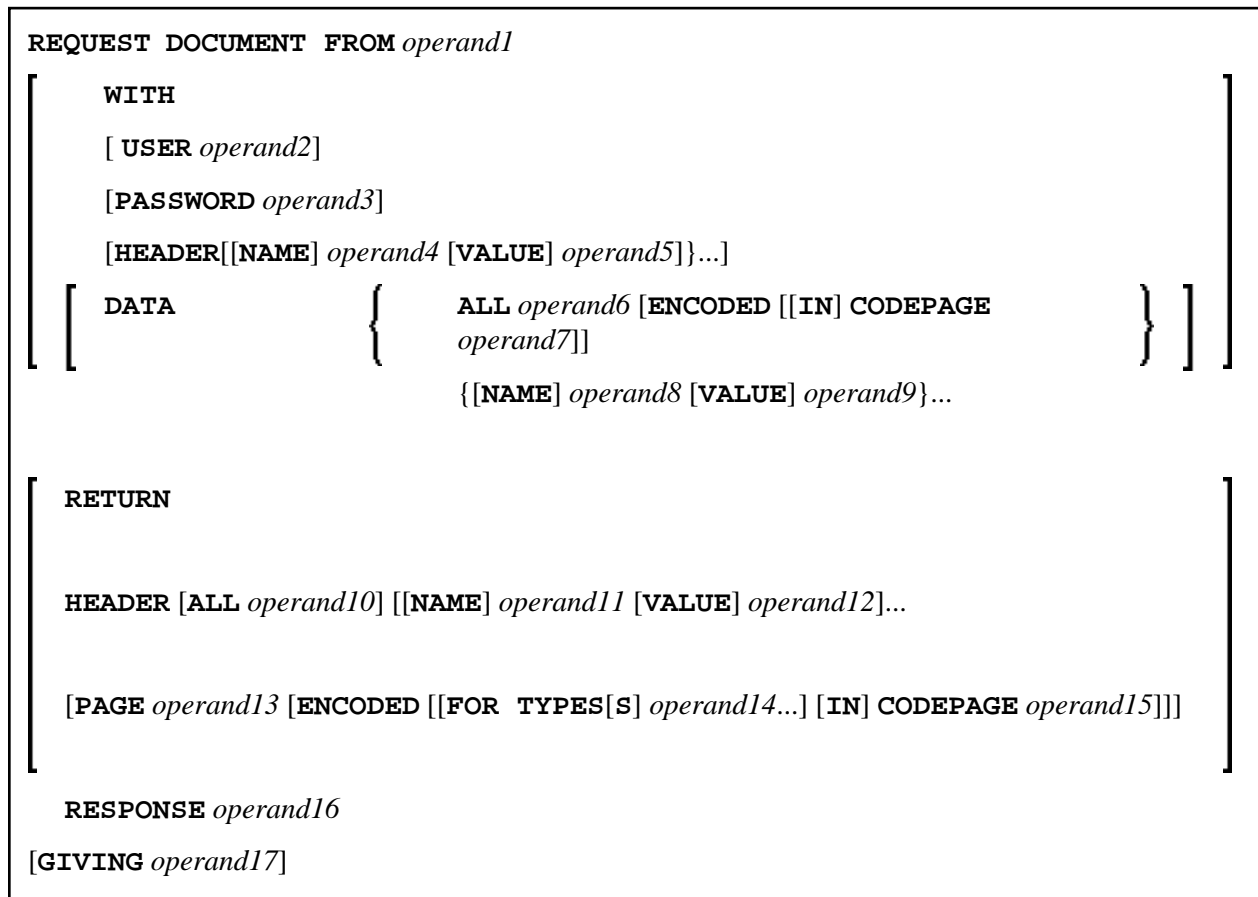


REQUEST DOCUMENT



Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Kodierung von eingehenden/ausgehenden Daten
- Beispiele

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Gehört zur Funktionsgruppe: *Internet und XML*

Funktion

Mit dem Statement `REQUEST DOCUMENT` haben Sie die Möglichkeit, auf ein externes System zuzugreifen.

Siehe auch *Statements für Internet- und XML-Zugang* im *Leitfaden zur Programmierung*.

Informationen bezüglich Unicode-Support finden Sie im Abschnitt *Statements* in der *Unicode and Code Page Support*-Dokumentation.

Einschränkungen für Cookies

Unter dem HTTP-Protokoll setzt ein Server Cookies ein, um Status-Informationen zur Client-Workstation zu verwalten.

REQUEST DOCUMENT wird mit Optionseinstellungen für das Internet implementiert. Dies bedeutet, dass abhängig von den Sicherheitseinstellungen Cookies verwendet werden.

Wenn in den Einstellungen der Internet-Optionen *Disabled* (Sperrern) gesetzt ist, werden keine Cookies versandt, auch wenn ein Cookie-Header (*operand4/operand5*) versandt wird. Benutzen Sie für Server-Umgebungen nicht die Internet-Optionseinstellung *Prompt* (Eingabeaufforderung). Mit dieser Einstellung bleibt der Server hängen, weil kein Client auf die Eingabeaufforderung reagieren kann.

In Großrechner-Umgebungen werden Cookies nicht unterstützt und ignoriert.

Cookies werden automatisch vom Windows-API verarbeitet. Das heißt: wenn Cookies im Browser aktiviert wurden, werden alle eingehenden Cookies gespeichert und mit der nächsten Anfrage automatisch versendet.

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur		Mögliche Formate												Referenzierung erlaubt	Dynam. Definition				
<i>operand1</i>	C	S					A												nein	ja
<i>operand2</i>	C	S					A												nein	ja
<i>operand3</i>	C	S					A												nein	ja
<i>operand4</i>	C	S					A												nein	ja
<i>operand5</i>	C	S					A	N	P	I	F	D	T	L					nein	ja
<i>operand6</i>	C	S					A	U	N	P	I	F	B	D	T	L			nein	ja
<i>operand7</i>	C	S					A												nein	ja
<i>operand8</i>	C	S					A												nein	ja
<i>operand9</i>	C	S					A	N	P	I	F	D	T	L					nein	ja
<i>operand10</i>		S					A												nein	ja
<i>operand11</i>	C	S					A												nein	ja
<i>operand12</i>		S					A	N	P	I	F	B	D	T	L				nein	ja
<i>operand13</i>		S					A	U				B							nein	ja
<i>operand14</i>	C	S					A												nein	ja
<i>operand15</i>	C	S					A												nein	ja
<i>operand16</i>		S									I4								nein	ja
<i>operand17</i>		S									I4								nein	nein

Syntax-Element-Beschreibung:

DOCUMENT FROM <i>operand1</i>	<p>Adresse des Dokuments:</p> <p><i>operand1</i> ist der URI, der zum Aufruf eines Dokuments benutzt wird.</p> <p>Anmerkung: Die folgenden Informationen gelten nur, wenn <i>operand1</i> mit <code>http://</code> oder <code>https://</code> beginnt.</p>
WITH	<p>WITH-Klausel:</p> <p>Sie können diese Klausel benutzen, um optional Benutzer/Passwort, Header und Einzelheiten zu den Daten für die Anforderung anzugeben.</p>
USER <i>operand2</i>	<p>Benutzer-Name:</p> <p><i>operand2</i> ist der Name des Benutzers, der für die Anforderung benutzt wird.</p>

<p>PASSWORD <i>operand3</i></p>	<p>Benutzer-Passwort:</p> <p><i>operand3</i> ist das Passwort des Benutzers, das für die Anforderung benutzt wird.</p>
<p>HEADER {[[NAME] <i>operand4</i> [VALUE] <i>operand5</i>]}...</p>	<p>HEADER-Klausel:</p> <p><i>operand4</i> und <i>operand5</i> können nur gemeinsam verwendet werden.</p> <ul style="list-style-type: none"> ● <i>operand4</i> ist der Name einer mit dieser Anforderung versandten HEADER-Variable. ● <i>operand5</i> ist der Wert einer mit dieser Anforderung versandten HEADER-Variable. <p>HEADER-Name für <i>operand4</i>:</p> <p>HEADER-Namen dürfen nicht CR/LF (Carriage Return/Line Feed) oder Doppelpunkt (:) enthalten. Dies wird nicht vom Statement REQUEST DOCUMENT überprüft. Gültige HEADER-Namen entnehmen Sie den HTTP-Spezifikationen. Aus Gründen der Kompatibilität mit der Web-Schnittstelle können Header-Namen mit Unterstrich (_) anstatt Bindestrich (-) geschrieben werden. (Intern wird der Unterstrich durch einen Bindestrich ersetzt).</p> <p>HEADER-Wert für <i>operand5</i>:</p> <p>HEADER-Werte dürfen nicht CR/LF enthalten. Dies wird vom Statement REQUEST DOCUMENT nicht überprüft. Gültige Header-Werte und -Formate entnehmen Sie den HTTP-Spezifikationen.</p> <p>Allgemeine Informationen zu Headers:</p> <p>Für eine HTTP-Anforderung sind einige Headers erforderlich, zum Beispiel: <i>Request-Method</i> oder <i>Content-Type</i>. Diese Headers werden automatisch erzeugt, und zwar abhängig von den mit dem Statement REQUEST DOCUMENT angegebenen Parametern.</p> <p>Siehe auch <i>Automatisch erzeugte Headers</i>.</p>
<p>DATA</p>	<p>DATA-Klausel:</p> <p>Sie können entweder einen spezifischen DATA-Variablennamen und -wert angeben (siehe <i>operand8</i> und <i>operand9</i> weiter unten) oder das vollständige Dokument (siehe <i>DATA ALL-Klausel</i> weiter unten).</p>

ALL <i>operand6</i>	<p><i>operand6</i> ist ein vollständiges, zu versendendes Dokument. Dieser Wert ist normalerweise erforderlich für die automatische HTTPAnforderungsmethode PUT (siehe <i>Automatisch erzeugte Headers</i>).</p> <p>Siehe <i>Kodierung von eingehenden/ausgehenden Daten, DATA ALL-Klausel</i>.</p>
[ENCODED [[IN] CODEPAGE <i>operand7]</i>	<p><i>operand6</i> wird von der voreingestellten (Default-)Codepage (Wert der Systemvariablen *CODEPAGE) in die in <i>operand7</i> angegebenen Codepage umkodiert.</p> <p>Siehe <i>Kodierung von eingehenden/ausgehenden Daten, DATA ALL-Klausel</i>.</p>

<p>{[NAME] <i>operand8</i> [VALUE] <i>operand9</i>}...</p>	<p>DATA-Variablenname und -wert:</p> <p><i>operand8</i> und <i>operand9</i> können nur gemeinsam benutzt werden:</p> <ul style="list-style-type: none"> • <i>operand8</i> ist der Name einer mit dieser Anfrage zu sendenden DATA-Variablen. Dieser Wert ist erforderlich für die HTTP-Anforderungsmethode POST (URL-Kodierung erforderlich, insbesondere Und-Zeichen (&), Gleichheitszeichen (=), Prozentzeichen (%). <p>Einschränkung:</p> <p>Wenn <i>operand8/operand9</i> angegeben wird und die Kommunikation standardmäßig <code>http://</code> oder <code>https://</code> ist, wird die Anfragemethode POST (siehe <i>Automatisch erzeugte Headers</i>) mit dem Inhaltstyp (<i>Content Type</i>) <code>application/x-www-form-urlencoded</code> benutzt.</p> <p>Im Verlauf der Anfrage werden <i>operand8/operand9</i> durch Gleichheitszeichen (=) und Und-Zeichen (&) voneinander getrennt. Deshalb dürfen die Operanden keine Und-Zeichen (&), Gleichheitszeichen (=) oder (wegen der URL-Kodierung) Prozentzeichen (%) enthalten. Diese Zeichen gelten als "unsicher" und müssen wie folgt kodiert werden:</p> <table border="1" data-bbox="722 1150 1377 1457"> <thead> <tr> <th>Zeichen</th> <th>URL-Kodierungssyntax</th> </tr> </thead> <tbody> <tr> <td>%</td> <td>%25</td> </tr> <tr> <td>&</td> <td>%26</td> </tr> <tr> <td>=</td> <td>%3D</td> </tr> </tbody> </table> <p>Siehe auch <i>Allgemeine Anmerkung zur URL-Kodierung</i>.</p>	Zeichen	URL-Kodierungssyntax	%	%25	&	%26	=	%3D
Zeichen	URL-Kodierungssyntax								
%	%25								
&	%26								
=	%3D								
<p>RETURN</p>	<p>RETURN-Klausel:</p> <p>Diese Klausel kann benutzt werden, um die HEADER- und/ oder PAGE-Rückgabeinformationen anzugeben.</p>								

<p>HEADER [ALL <i>operand10</i>]</p>	<p>RETURN HEADER ALL-Klausel:</p> <p>Wenn diese Klausel angegeben wird, enthält <i>operand10</i> alle mit der HTTP-Rückmeldung angegebenen Header-Werte.</p> <p>Die erste Zeile enthält die Status-Informationen, und alle folgenden Zeilen enthalten die Headers als Paare mit Namen und Werten. Die Namen enden immer auf einen Doppelpunkt (:), und die Werte enden mit einem Zeilenvorschub (LF). (Intern werden alle CR/LF auf Zeilenvorschub, d.h. LF, umgesetzt.)</p>
<p>HEADER [[NAME] <i>operand11</i>] [VALUE] <i>operand12</i>]...</p>	<p>RETURN HEADER NAME/VALUE-Klausel:</p> <p>Wenn diese Klausel angegeben wird, werden nur spezifische HEADER-Informationen zurückgegeben.</p> <p><i>operand11</i> und <i>operand12</i> können nur in Verbindung miteinander benutzt werden:</p> <ul style="list-style-type: none"> • <i>operand11</i> ist der Name eines in dieser Anfrage erhaltenen Headers. Die HEADER-Angabe ist für HTTP erforderlich. • <i>operand12</i> ist der Wert eines in dieser Anfrage erhaltenen Headers. Die HEADER-Angabe ist für HTTP erforderlich. <p>Rückgabe des Header-Namens für <i>operand11</i>:</p> <p>Aus Gründen der Kompatibilität mit dem Natural Web Interface können Header-Namen mit Unterstrich (_) anstatt Bindestrich (-) geschrieben werden. Intern wird der Unterstrich durch einen Bindestrich ersetzt.</p> <p>Wenn <i>operand11</i> eine Leerzeichenkette ist, werden die Status-Informationen zurückgegeben:</p> <p>HTTP/1.0 200 OK</p>
<p>RETURN PAGE</p>	<p>RETURN PAGE-Klausel:</p> <p>Sie können die PAGE-Klausel benutzen, wenn Sie möchten, dass die eingehenden Daten in einer spezifischen Codepage kodiert werden.</p> <p>Siehe <i>Kodierung von eingehenden/ausgehenden Daten</i>, <i>RETURN PAGE-Klausel</i> weiter unten.</p>
<p>PAGE <i>operand13</i></p>	<p><i>operand13</i> ist das für diese Anfrage zurückgegebene Dokument.</p> <p>Siehe <i>Kodierung von eingehenden/ausgehenden Daten</i>, <i>RETURN PAGE-Klausel</i> weiter unten.</p>

<p>[ENCODED [[FOR TYPE[S] operand14...] [IN] CODEPAGE operand15]]</p>	<p><i>operand14</i> ist die Liste der Mime-Typen, für die eine Kodierung des zurückgegebenen Dokuments in <i>operand13</i> ausgeführt wird.</p> <p>Siehe <i>Kodierung von eingehenden/ausgehenden Daten, RETURN PAGE-Klausel</i> weiter unten.</p> <p><i>operand15</i> ist die Codepage, die erforderlichenfalls für die Kodierung von <i>operand13</i> benutzt wird.</p> <p>Wenn der Wert von <i>operator15</i> leer ist, wird keine Konvertierung vorgenommen. <i>operand13</i> ist dann mit der Standard-Codepage kodiert (Profilparameter CP im Configuration Utility).</p> <p>Siehe <i>Kodierung von eingehenden/ausgehenden Daten, RETURN PAGE-Klausel</i> weiter unten.</p>
<p>RESPONSE operand16</p>	<p>RESPONSE-Klausel:</p> <p>Geben Sie die RESPONSE-Klausel an, wenn Sie möchten, dass die Response-Codenummer der Anforderung angezeigt wird.</p> <p><i>operand16</i> ist die Response-Codenummer der Anforderung, zum Beispiel: 200 (Anforderung erledigt).</p> <p>Siehe auch <i>Übersicht über Response-Nummern für HTTP/HTTPS-Anfragen</i>.</p>
<p>GIVING operand17</p>	<p>GIVING-Klausel:</p> <p><i>operand17</i> enthält den Natural-Fehler, wenn die Anforderung nicht ausgeführt werden konnte.</p>

Automatisch erzeugte Header (operand4/5)

Request-Method

Die folgenden Werte werden für *operand5* unterstützt: HEAD, POST, GET und PUT.

Die folgende Tabelle zeigt die automatische Berechnung der *request-method* in Abhängigkeit von den vorgegebenen Operanden:

Operand	Request-Method			
	HEAD	POST	GET	PUT
WITH HEADER (<i>operand4/operand5</i>)	optional	optional	optional	optional
WITH DATA (<i>operand7/operand8</i>)	nicht angegeben	angegeben	nicht angegeben	nur bei Option ALL (<i>operand6</i>)
RETURN HEADER (<i>operand10 bis operand12</i>)	angegeben	angegeben	optional	optional
RETURN PAGE (<i>operand13</i>)	nicht angegeben	angegeben	angegeben	optional

Content-Type

Wenn die *request-method* POST ist, muss ein Header des Typs *content-type* mit der HTTP-Anfrage angegeben werden. Wenn kein *content-type* explizit gesetzt wird, ist der automatisch generierte Wert von *operand5* wie folgt:

```
application/x-www-form-urlencoded
```

Anmerkung:

Es ist möglich, die automatisch erzeugten Headers zu überschreiben. Natural überprüft sie nicht auf Fehler. Es können unerwartete Fehler auftreten.

Allgemeiner Hinweis zur URL-Kodierung

Wenn Sie POST-Daten mit dem Inhaltstyp `application/x-www-form-urlencoded` versenden, müssen bestimmte Zeichen mittels URL-Kodierungen dargestellt werden, was bedeutet, dass das Zeichen durch einen hexadezimalen Zeichencode (`%hexadecimal-character-code`) ersetzt wird. Die vollständigen Einzelheiten, wann und warum die URL-Kodierung erforderlich ist, sind in RFC 1630, RFC 1738 und RFC 1808 erläutert. Sie finden hier einige grundsätzliche Informationen. Alle Nicht-ASCII-Zeichen (d.h. gültige ISO 8859/1-Zeichen, die nicht auch ASCII-Zeichen sind) müssen URL-kodiert sein, z.B. die Datei `köln.html` würde in einer URL als `k%F6ln.html` erscheinen.

Einige Zeichen gelten als "unsicher", wenn Web-Seiten per Email angefordert werden.

Diese Zeichen lauten:

Zeichen	URL-Kodierungssyntax
Tab-Zeichen	%09
Leerzeichen	%20
[%5B
\	%5C
]	%5D
^	%5E
‘	%60
{	%7B
	%7C
}	%7D
~	%7E

Wenn Sie URLs schreiben, sollten Sie diese Zeichen URL-kodieren.

Einige Zeichen haben spezielle Bedeutungen in URLs, wie z.B. der Doppelpunkt (:), der das URL-Schema vom Rest des URLs abtrennt, der doppelte Schrägstrich (//), der angibt, dass der URL der Common Internet Scheme-Syntax entspricht, und das Prozentzeichen (%). Wenn diese Zeichen als Teile von Dateinamen erscheinen, müssen sie generell URL-kodiert werden, um sie von ihrer Sonderbedeutung in URLs zu unterscheiden (dies ist eine Vereinfachung, die vollständigen Informationen finden Sie in den RFCs).

Diese Zeichen sind:

Zeichen	URL-Kodierungssyntax
"	%22
#	%23
%	%25
&	%26
+	%2B
,	%2C
/	%2F
:	%3A
<	%3C
=	%3D
>	%3E
?	%3F
@	%40

Übersicht über Response-Nummern für HTTP-Anforderungen

Status	Wert	Rückmeldung
STATUS CONTINUE	100	OK — Fortfahren mit der Anforderung.
STATUS SWITCH_PROTOCOLS	101	Server hat die Protokolle im Upgrade-Header geändert.
STATUS OK	200	Anforderung erledigt.
STATUS CREATED	201	Objekt erstellt, Grund = neuer URL.
STATUS ACCEPTED	202	Asynchrone Beendigung (TBS).
STATUS PARTIAL	203	Teilweise Beendigung.
STATUS NO_CONTENT	204	Keine Infos zurückzugeben.
STATUS RESET_CONTENT	205	Anforderung abgeschlossen, aber Inhalt zurückgesetzt.
STATUS PARTIAL_CONTENT	206	Teilweises GET vollendet.
STATUS AMBIGUOUS	300	Server konnte keine Entscheidung über Rückmeldung fällen.
STATUS MOVED	301	Objekt permanent übertragen.
STATUS REDIRECT	302	Objekt zeitweise übertragen.
STATUS REDIRECT_METHOD	303	Umleiten ohne neue Zugriffsmethode.
STATUS NOT_MODIFIED	304	Wenn geändert — seit wann nicht geändert.
STATUS USE_PROXY	305	Umleiten zu Proxy, Adress-Header gibt zu benutzende Proxy an.

Status	Wert	Rückmeldung
STATUS REDIRECT_KEEP_VERB	307	HTTP/1.1: dasselbe Verb beibehalten.
STATUS BAD_REQUEST	400	Ungültige Syntax.
STATUS DENIED	401	Zugriff verweigert.
STATUS PAYMENT_REQ	402	Zahlung erforderlich.
STATUS FORBIDDEN	403	Anforderung nicht erlaubt.
STATUS NOT_FOUND	404	Objekt nicht gefunden.
STATUS BAD_METHOD	405	Methode ist nicht zulässig.
STATUS NONE_ACCEPTABLE	406	Keine Rückmeldung für gefundenen Client annehmbar.
STATUS PROXY_AUTH_REQ	407	Proxy-Echtheitsprüfung erforderlich.
STATUS REQUEST_TIMEOUT	408	Server-Zeitüberschreitung – warten auf Anforderung.
STATUS CONFLICT	409	Benutzer sollte mit mehr Informationen neu starten.
STATUS GONE	410	Die Ressource steht nicht mehr zur Verfügung.
STATUS LENGTH_REQUIRED	411	Der Server verweigerte die Annahme der Anforderung ohne Länge.
STATUS PRECOND_FAILED	412	In Anfrage angegebene Vorbedingung unzulässig
STATUS REQUEST_TOO_LARGE	413	Anforderungselement war zu groß.
STATUS URL_TOO_LONG	414	Anforderungs-URL zu lang.
STATUS UNSUPPORTED_MEDIA	415	Nicht unterstützter Medien-Typ.
STATUS SERVER_ERROR	500	Interner Server-Fehler.
STATUS NOT_SUPPORTED	501	"Required" nicht unterstützt.
STATUS BAD_GATEWAY	502	Fehler-Rückmeldung vom Gateway.
STATUS SERVICE_UNAVAIL	503	Zeitweise überlastet.
STATUS GATEWAY_TIMEOUT	504	Zeitüberschreitung – warten auf Gateway.
STATUS VERSION_NOT_SUP	505	HTTP-Version nicht unterstützt.

Response 301 - 303 (Umleitung)

Umleitung bedeutet, dass der angeforderte URL umgezogen ist. Als Response (Rückmeldung) wird der Rückgabe-Header mit dem Namen LOCATION (Adresse) angezeigt. Dieser Header enthält den URL, wohin die angeforderte Seite umgezogen ist. Eine neue REQUEST DOCUMENT-Anfrage kann benutzt werden, um die umgezogene Seite zu suchen.

HTTP-Browser leiten automatisch zur neuen URL um, aber das Statement `REQUEST DOCUMENT` nimmt die Umleitung nicht automatisch vor.

Response 401 (Verweigert)

Die Rückmeldung `Access Denied` (Zugriff verweigert) bedeutet, dass die angeforderte Seite nur aufgerufen werden kann, wenn mit der Anfrage eine gültige Benutzer-ID und ein gültiges Passwort angegeben werden. Als Rückmeldung wird der Rückgabe-Header mit dem Namen `WWW-AUTHENTICATE` mit dem für diese Anfrage erforderlichen Bereich ausgegeben.

HTTP-Browser zeigen normalerweise einen Dialog mit Benutzer-ID und Passwort an, aber beim Statement `REQUEST DOCUMENT` wird kein Dialog angezeigt.

Kodierung von eingehenden/ausgehenden Daten

Bei der Datenübertragung mit dem Statement `REQUEST DOCUMENT` kommt es normalerweise nicht zu Konvertierungen von Codepages. Wenn Sie möchten, dass die ausgehenden und/oder eingehenden Daten in einer bestimmten Codepage kodiert werden, können sie die Klausel `DATA ALL` und/oder die Klausel `RETURN PAGE` verwenden, um dies anzugeben.

DATA ALL-Klausel

Zur Kodierung der ausgehenden Daten wird die Klausel `DATA ALL` benutzt:

```
ALL operand6 [ENCODED [[IN] CODEPAGE operand7]]
```

Syntax-Beschreibung:

ALL operand6	operand6 ist ein vollständiges Dokument, das versandt werden soll. Dieser Wert wird normalerweise für die automatische HTTP-Anforderungsmethode <code>PUT</code> benötigt (siehe <i>Automatisch erzeugte Headers</i>).
[ENCODED [[IN] CODEPAGE operand7]]	operand6 wird von der voreingestellten (Default-)Codepage (Wert der Systemvariablen <code>*CODEPAGE</code>) in die in operand7 angegebenen Codepage umgesetzt.

RETURN PAGE-Klausel

Zur Kodierung von eingehenden Daten wird die Klausel `RETURN PAGE` verwendet:

```
[PAGE operand13 [ENCODED [[FOR TYPE[S] operand14...] [IN] CODEPAGE operand15]]]
```

Als Rückmeldung für eine HTTP-/HTTPS-Anfrage können eingehende Daten binäre Daten (zum Beispiel `image/gif`) oder Zeichen-Daten (zum Beispiel `text/html`) enthalten. Zusammen mit der Rückmeldung erhält das Statement `REQUEST DOCUMENT` einen Parameter, der die Art des Inhalts des angeforderten Dokuments angibt (Mime-Typ). Dieser Parameter kann Informationen über die Codepage enthalten, in der das Dokument kodiert ist.

Diese Klausel bietet eine automatische Umsetzung in die voreingestellte (Default-)Codepage (Wert der Systemvariablen *CODEPAGE) der Natural-Session.

Syntax-Beschreibung:

RETURN PAGE <i>operand13</i>	Es erfolgt keine Kodierung der zurückgegebenen Seite; das bedeutet, die Seite bleibt so kodiert, wie sie vom HTTP-Server geliefert wird.
RETURN PAGE <i>operand13</i> ENCODED	Wenn der zurückgegebene Mime-Typ eine Kodierung enthält, dann wird <i>operand13</i> von dieser Codepage in die voreingestellte (Default-)Codepage (A/B) oder (U) umkodiert. Siehe Anmerkung unten.
RETURN PAGE <i>operand13</i> ENCODED [IN] CODEPAGE <i>operand15</i>	Wenn der der zurückgegebene Mime-Typ keine Kodierung enthält, wird <i>operand13</i> von der mit <i>operand15</i> definierten Codepage in die voreingestellte (Default-)Codepage (Wert der Systemvariablen *CODEPAGE) (A/B) oder (U) umkodiert.
RETURN PAGE <i>operand13</i> [ENCODED [[FOR TYPE[S] <i>operand14...</i>] [IN] CODEPAGE <i>operand15</i>]]	Wenn der zurückgegebene Mime-Typ keine Codierung enthält, wird eine zusätzliche Prüfung durchgeführt, ob der zurückgegebene Mime-Typ mit einem der in <i>operand14</i> gelieferten Typen übereinstimmt. Wenn eine Übereinstimmung vorliegt, wird <i>operand13</i> von der mit <i>operand15</i> definierten Codepage in die voreingestellte (Default-)Codepage (A/B) oder (U) umkodiert.

Anmerkung:

"Zurückgegebener Mime-Typ enthält eine Kodierung" bedeutet, dass der HTTP-Server einen Content-Type-Header mit einer charset=-Klausel zurückliefert, zum Beispiel:
charset=ISO-8859-1.

Beispiele

- Beispiel 1 — Allgemeine Anforderung
- Beispiel 2 — Einfache Get-Anforderung (keine Daten)
- Beispiel 3 — Einfache Head-Anforderung (keine zurückgelieferte Seite)
- Beispiel 4 — Einfache Post-Anforderung (Voreinstellung)
- Beispiel 5 — Einfache Put-Anforderung (mit allen Daten)

Anmerkung:

Es gibt einen Beispiel-Dialog V5-RDOC für dieses Statement in der Beispiel-Library SYSEXV.

Beispiel 1 — Allgemeine Anforderung

```
REQUEST DOCUMENT FROM "http://bolsap1:5555/invoke/sap.demo/handle_RFC_XML_POST"
WITH
  USER #User PASSWORD #Password
  DATA
  NAME 'XMLData'          VALUE #Queryxml
  NAME 'repServerName'   VALUE 'NT2'
RETURN
  PAGE #Resultxml
RESPONSE #rc
```

Beispiel 2 — Einfache Get-Anforderung (keine Daten)

```
REQUEST DOCUMENT FROM "http://pcnatweb:8080"
RETURN
  PAGE #Resultxml
RESPONSE #rc
```

Beispiel 3 — Einfache Head-Anforderung (keine zurückgelieferte Seite)

```
REQUEST DOCUMENT FROM "http://pcnatweb"
RESPONSE #rc
```

Beispiel 4 — Einfache Post-Anforderung (Voreinstellung)

```
REQUEST DOCUMENT FROM "http://pcnatweb/cgi-bin/nwwcgi.exe/sysweb/nat-env"
WITH
  DATA
  NAME 'XMLData'          VALUE #Queryxml
  NAME 'repServerName'   VALUE 'NT2'
RETURN
  PAGE #Resultxml
RESPONSE #rc
```

Beispiel 5 — Einfache Put-Anforderung (mit allen Daten)

```
REQUEST DOCUMENT FROM "http://pcnatweb/test.txt"
WITH
  DATA ALL      #document
RETURN
  PAGE #Resultxml
RESPONSE #rc
```