SupportPac CA8J: Generate Basic Authentication headers for a CICS HTTP client

Version 1.0

8 November 2006

Peter Havercan
IBM United Kingdom Ltd
MP208 Hursley Park
Winchester, Hampshire, UK
SO21 2JN

peter_havercan@uk.ibm.com

Take Note!

Before using this report be sure to read the general information under "Notices".

First Edition, November 2006

This edition applies to Version 1.0 of SupportPac CA8J and to all subsequent releases and modifications unless otherwise indicated in new editions.

Table of Contents

Notices

The following paragraph does not apply in any country where such provisions are inconsistent with local law.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of the intellectual property rights may be used instead of the IBM product.

Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA.

The information contained in this document has not be submitted to any formal IBM test and is distributed AS-IS. The use of the information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Trademarks and service marks

The following terms, used in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- CICS
- CICS Transaction Server
- z/OS

Summary of Amendments

Date	Changes
8 November 2006	Initial release

Preface

This SupportPac demonstrates how you can customize CICS Transaction Server for z/OS to provide HTTP Basic Authentication headers when CICS is being used as an HTTP client. This support is required when an HTTP server that is being accessed by CICS requires Basic Authentication credentials before it can be accessed. Examples are given for customizing CICS Transaction Server V2.3, when the HTTP client support uses the DFHWBCLI utility program, for the "SOAP for CICS" feature, based on CICS TS V2.3, and for CICS Transaction Server V3.1, when the HTTP client support is based on the EXEC CICS WEB programming interface,

Bibliography

- *HTTP Authentication: Basic and Digest Access Authentication,* RFC2617
  (at <http://www.ietf.org/rfc/rfc2617.txt>)

- *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies,* RFC1521
  (at <http://www.ietf.org/rfc/rfc1521.txt>)

Chapter 1. Introduction

Introduction to HTTP Basic Authentication

The HTTP protocol allows server systems to request a simple authentication challenge in which the HTTP client is required to provide a username and password. The authentication system is described in *HTTP Authentication: Basic and Digest Access Authentication,* RFC2617. The HTTP Digest Authentication protocol is also described there, but that protocol is **not** being addressed by this SupportPac.

In the Basic Authentication protocol, the HTTP server system examines the incoming HTTP headers for an Authorization header in the following format:

```
Authorization: Basic xxxxxxxxxxxx
```

where *xxxxxxxxxxxx* is a base64-encoded form of the string `username:password`, which contains the client user's credentials (in an ASCII codepage). Base-64 encoding is described in RFC1521. **Credentials**, in this context, just means the client's username and password.

If the HTTP client has not provided an Authorization header, or has provided one with incorrect credentials, the HTTP server rejects the request with an HTTP 401 response which includes the response header:

```
WWW-Authenticate: Basic realm="realmname"
```

Where *realmname* is the name of a security realm in which a set of credentials can be validated. It is up to the server to determine the meaning and scope of the realm.

Problems with implementing Basic Authentication in a CICS application

Although CICS does provide mechanisms for sending HTTP headers from an HTTP client system, the format of the Authorization header means that it is quite difficult to compose one from a typical CICS Cobol application. This is because:

- The base-64 algorithm requires that the credential string being encoded has to be decomposed into fragments that are six bits wide. Each six-bit fragment is then encoded into an eight-bit character in the base-64 alphabet. This is quite difficult to do in Cobol.

- The credential string itself must be transformed into ASCII before being encoded into base-64. This is different from all other EBCDIC to ASCII conversions, which are handled automatically by CICS just before the data is written to the socket. Codepage conversion is also quite difficult to handle in Cobol.

The above two problems are addressed by this SupportPac. There is also a problem in determining the correct credentials to be applied to a particular server, or a particular realm within a server, and in safely applying them in an application without unnecessarily exposing them. These latter two problems are not addressed by this SupportPac.

Chapter 2. SupportPac contents

What this SupportPac contains

The components contained in this SupportPac are as follows:

ca8j.pdf A PDF file containing this User guide

ca8j.zip A zip file containing the following:

license	A folder that contains your license for this SupportPac, in a number of national languages.
ca8j.load.pds	An unloaded copy of CA8J.LOAD, which contains the load modules BASE64 and HTTPAUTH in a partitioned dataset load library.
ca8j.source.pds	An unloaded copy of CA8J.SOURCE, which contains the source components of this SupportPac in a partitioned dataset.
asmjcl.jcl	The JCL used to assemble and link BASE64 and HTTPAUTH.
cobjcl.jcl	The JCL used to translate, compile, and link BASIC230, DFH$WSSB, BASIC310, WSBASIC, WSSAMPLE
csddefns.txt	An input stream for the DFHCSDUP utility to define sample RDO definitions for use by the example programs.
base64.asm	Assembler source for BASE64, the base-64 encode and decode routine.
httpauth.asm	Assembler source for HTTPAUTH, a routine to construct an HTTP Authorization header.
basic230.cob	Cobol source for a Basic Authentication sample for use on CICS/TS 2.3
basic310.cob	Cobol source for a Basic Authentication sample for use on CICS/TS 3.1
dfh$wssb.cob	A modified version of the "Soap for CICS" sample program with enhancements to demonstrate the use of Basic Authentication.
s4cbasic.cob	Cobol source for a code fragment for use with the "Soap for CICS" feature sample DFH$WSSB.
s4cbasix.cob	A pre-translated version of **s4cbasic.cob**.
s4cbadef.cob	A Cobol copybook of definitions required by the **s4cbasic.cob** fragment.
s4cbalnk.cob	A Cobol copybook for the Linkage Section definition required by the **s4cbasic.cob** fragment.
wsbasic.cob	A CICS Web Services pipeline Transport Handler that is capable of producing Basic Authentication headers
wssample.cob	A simple SOAP requester application that initiates Basic Authentication request.
license	A folder that contains your license for this SupportPac, in a number of national languages.

Chapter 3. How to install the SupportPac

To use the programs in this SupportPac, first unzip the main file **ca8j.zip** into a convenient folder on your workstation. Then use an FTP program to transfer the two files **ca8j.load.pds** and **ca8j.source.pds** to sequential files on your MVS or z/OS system. Be sure to use the FTP BINARY option to transfer both of these files without conversion. They should be allocated with attributes RECFM=FB, BLKSIZE=3120, LRECL=80. These two files are both unloaded PDS data sets. You must reload them as PDS files before use, using the TSO RECEIVE command:

```
RECEIVE INDSNAME(ca8j.load.pds)
RECEIVE INDSNAME(ca8j.source.pds)
```

These commands should create the partitioned datasets CA8J.LOAD and CA8J.SOURCE. The contents of CA8J.SOURCE are identical to the source files base64.asm, basic230.cob, basic310.cob, etc., so you may not need to transfer these files to your MVS system.

Chapter 4. How to use the examples in the SupportPac

Example for CICS Transaction Server V2.3

If you are using CICS Transaction Server V2.3 you can use the BASIC230 example. This is a simple HTTP client program that sends an HTTP GET request to a target server. You must modify the target URL in the variable TARGET-URL, and you must modify the username and password in the variables HTTPAUTH-USERID and HTTPAUTH-PASSWORD.

The sample program links to HTTPAUTH to create an Authorization header, and adds it to the list of headers that are send by the DFHWBCLI utility program.

Example for the "SOAP for CICS" feature

If you are using the "SOAP for CICS" feature, you can use the modified DFH$WSSB program that comes with this SupportPac. This is a modified version of the DFH$WSSB sample that came with the feature. (The modifications are highlighted by the identifier CA8J in columns 1-4.)

You should modify the instructions that move the username and password (and their lengths) to the variables HTTPAUTH-USERID, HTTPAUTH-PASSWORD, HTTPAUTH-USERID-LENGTH, and HTTPAUTH-PASSWORD-LENGTH.

The sample program uses the copybooks S4CBADEF, S4CBASIX to cause linkage to HTTPAUTH to create an Authorization header. If you use the Integrated compiler, you should use the S4CBASIC copybook instead of S4CBASIX.

This sample works by appending the Authorization header to a SoapAction header that is usually sent by the "SOAP for CICS" feature.

Example for CICS Transaction Server V 3.1

If you are using CICS Transaction Server V3.1 for plain HTTP requests (not Web Services) you can use the BASIC310 example. You must customize the variables HTTP-HOST and HTTP-PATH to specify the target URL to which the request must be sent, and the variables HTTPAUTH-USERID-LENGTH, HTTPAUTH-PASSWORD-LENGTH, HTTPAUTH-USERID, and HTTPAUTH-PASSWORD to supply the username and password that you want to send.

The sample program links to HTTPAUTH to create the Authorization header, and then sends it using the EXEC WEB WRITE HTTPHEADER command and the EXEC CICS WEB SEND command.

Example for CICS Transaction Server V3.1 Web Services

If you are using CICS Transaction Server V3.1 Web Services support you can use the WSSAMPLE example to initiate a Web Service request pipeline. You should customize the target Web Service URL in the WEBSERVICE-URL variable. This initiates a Web Service request on the WSBASIC pipeline, for which a sample definition is provided in CSDDEFNS in CA8J.SOURCE. This pipeline definition points to a Web Service configuration XML file. An example of this is provided in WSBASIC2. You should copy this to an HFS file of your choice and then modify the WSBASIC pipeline definition to specify the location to which you copy it. The XML configuration file specifies a **transport handler** of WSBASIC, and it is here that the Basic Authentication header creation is done. You should customize the variable HTTPAUTH-USERID-LENGTH, HTTPAUTH-PASSWORD-LENGTH, HTTPAUTH-USERID, and HTTPAUTH-PASSWORD to provide the username and password that you want to be sent.

The WSBASIC transport handler uses a similar technique to the "SOAP for CICS" example to append the Authorization header to a SoapAction header. The pipeline itself does not provide a mechanism for adding arbitrary headers to the SOAP request.

-- End of Document --