Interactive SMSC

HTTP Connections (MT)

18/08/2011
v1.0

Table of Contents

Input protocol

Protocol	
URL	http://clients.actelme.com/MainSMS/push_Content.aspx

HTTP Parameters		
ASPX Page	push_Content.aspx	
Fixed Input ***(Please Find on the last page all the required fixed details)***		
Dynamic Input		
SMS Prameters		
Receiver	Mobile number	Integer
Originator	Short code	Integer
Countryname	Country name **(Check the below list)**	Alphabetic String
Operatorname	Operator name **(Check the below list)**	Alphabetic String
ContentBody	Message Content	String
ContentTypeDetails	• Unicode for UCS2 (Arabic) • Text for English • Wap for wap content	Alphabetic String
ContentSubType	• Smsreply (single sms reply) • Smsbulk (Bulk campaign) • Smsmtbilling (single MT billing reply) • Wapreply (Wap reply) • Wapbulk (Bulk campaign) • Wapmtbilling (single Wap MT billing reply)	Alphabetic String
username	**XXXX**	Alphanumeric String
Password	**XXXX**	Alphanumeric String
Contentid	Unique ID (SQL Based) EG : 06A839EF-6E7C-4D2F-B222-84EFB44EDF60 if possible on each process or normal unique id	Alphanumeric String

Return codes

Error Table		
Returned Values	**Description**	**Type**
Invalid Originator	Invalid short code	Alphabetic String
Invalid Receiver	Invalid mobile number	Alphabetic String
Invalid Countryname	Invalid or empty country name	Alphabetic String
Invalid Operatorname	Invalid or empty operator name	Alphabetic String
Invalid ContentBody	Empty message or invalid text	Alphabetic String
Invalid ContentTypeDetails	Empty ContentTypeDetails	Alphabetic String
Invalid ContentSubType	Empty ContentSubType	Alphabetic String
Invalid Contentid	Empty ID	Alphabetic String
Error Posting	Internal link error	Alphabetic String
ok	Message Sent Successfully	Alphabetic String

Requirements
1. *For Arabic sms we support ucs2 characters. (Hex Code)* ***2.*** *For English sms we support plain latin characters.*

Output
The result is either starts with: ***Invalid (error) with error description*** or **ok** as keyword

Samples

Example	
Text Request	http://clients.actelme.com/MainSMS/push_Content.aspx?username=XXXX&password=YYYY&receiver=ZZZZZZZ&ContentBody=test%20SMS&originator=AAAA&countryname=ksa&operatorname=zain&Contentid=AA814898-0CDA-48F6-BD22-1D4CB8ED2C9B&ContentTypeDetails=text&ContentSubType=smsreply
Unicode Request	http://clients.actelme.com/MainSMS/push_Content.aspx?username=XXXX&password=YYYY&receiver=ZZZZZZ&ContentBody=066106620663&originator=AAAA&countryname=kSA&operatorname=zain&Contentid= AA814898-0CDA-48F6-BD221D4CB8ED2C9B&ContentTypeDetails=Unicode&ContentSubType=smsreply
Wap Request	http://clients.actelme.com/MainSMS/push_Content.aspx?username=XXXX&password=YYYY&receiver=ZZZZZZ&ContentBody=Http://www.google.com&originator=AAAA&countryname=ksa&operatorname=zain&Contentid=AA814898-0CDA-48F6-BD22-1D4CB8ED2C9B&ContentTypeDetails=wap&ContentSubType=wapreply
Bulk Request	http://clients.actelme.com/MainSMS/push_Content.aspx?Receiver=XXXXX&Originator=XXX&username=XXXX&Password=YYYY&ContentSubType=smsbulk&ContentBody=test%20bulk&ContentTypeDetails=text&contentid=AA814898-0CDA-48F6-BD22-1D4CB8ED2C9B

Annexes

Country	Operator
Algeria	Djezzy
Algeria	Mobilis
Algeria	Nedjma
Bahrain	batelco
Bahrain	Zain
Egypt	Mobinil
Egypt	Vodafone
Egypt	Etisalat
International	Smartcom
Iran	MTN
Iraq	ZainIQ
Iraq	Etisaluna
Iraq	Korectel
Iraq	asiacell
Iraq	Iraqna
Iraq	ZainIQ
Jordan	Jordan Telecom
Jordan	Zain
Jordan	Orange
Jordan	Umniah

Jordan	Xpress
KSA	MOBILY
KSA	STC
KSA	Zain
kuwait	viva
kuwait	Zain
kuwait	wataniya
Lebanon	Alfa
Lebanon	MTCTouch
Libya	Libyana
Libya	elmadar
Morocco	Wana
Morocco	MAROCTEL
Morocco	MEDITEL
Oman	OmanTel
Oman	Nawras
Oman	omanMobile
Palestine	Jawal
Palestine	Zain
Palestine	Paltel
Qatar	qtel
Spain	Europe
Sudan	MTN
Sudan	Sudatel

Sudan	Zain
Syria	MTN
Syria	syriatel
Tunisia	Tunisiana
Tunisia	Tunistelecom
UAE	DU
UAE	Etisalat
yemen	Y
yemen	sabafon
yemen	MTN
yemen	yemenMobile

Annex 2 – Glossary

Aggregators/mobile messaging aggregators: Entities that provide access to mobile operators. Often referred to as service providers, an aggregator connects to the mobile operators and though one connection to its clients, aggregates access to multiple operators.

Gateway/messaging gateway: This is a system that handles receiving and sending messages between other messaging entities. They provide the single point of access and implement the notions of queuing, retry failed messages and implementation of required communication protocols.

Large scale competition: Large scale competitions, often referred to as super promotions, are services that Actel specializes in. Large scale competitions are by definitions services accessible by a large user base and generate big amounts of traffic.

VAS Promotion Systems: The system that implement the large scale competitions. VAS (Value Added Service) Promotion Systems refers to the application layer that Actel implements.

MO: Mobile Originating messages are messages sent by the user.

MT: Mobile Terminating messages are messages sent by Actel to the user.

Dialogue/MO-MT Dialogue: Represents a text based SMS dialogue that Actel's Systems holds with a user. A user sends and MO, Actel answers with an MT triggering the user to send another MO, thus establishing a dialogue with the user.

Base Bulk: Broadcasts to subscribers of the operators who have not yet joined the promotion.

Teaser Bulks: Broadcasts to subscribers who have joined the promotion and are being invited to play some more.

MO Inbound: The MO traffic that is sent to ACTEL by any peer.

MO Outbound: The MO Traffic that is sent from ACTEL to any other peer.

MT Inbound: The MT Traffic that is received by ACTEL from any peer.

MT Outbound: The MT traffic that is sent by ACTEL to any peer.

Submission, Hit, Transmission: These terms designate a single HTTP PDU sent from or to ACTEL

Annex 3 – MD5 encryption function

DISCLAIMER: Please be advised that the provided code samples are provided for your convenience. Use them at your own risk.

VB.Net

```
Public Function MD5Encrypt(ByVal EncString As String) As String
        'Variable Declarations
        Dim MD5String As String
        Dim EncStringBytes() As Byte
        Dim Encoder As New UTF8Encoding
        Dim MD5Hasher As New MD5CryptoServiceProvider
        'Converts the String to bytes
        EncStringBytes = Encoder.GetBytes(EncString)
        'Generates the MD5 Byte Array
        EncStringBytes = MD5Hasher.ComputeHash(EncStringBytes)
        'Create MD5 hash
        MD5String = BitConverter.ToString(EncStringBytes)
        MD5String = MD5String.Replace("-", "")
        'Returns the MD5 encrypted string to the calling object
        Return MD5String
    End Function
```

Usage

Signature=MD5Encrypt("ClientPassword@" & smssender)

PHP

```
$Signature= md5("clientpassword@".$smssender);
```

Java

```
import java.security.MessageDigest;

public class Main {
   public static void main(String[] args) {
      try{
         MessageDigest digest = java.security.MessageDigest.getInstance("MD5");
         //Put in here the String that you want to has its MD5 sig
         String sig = new String("actel");
         digest.update(sig.getBytes());
         byte[] hash = digest.digest();
         String result = "";
         for (int i=0; i < hash.length; i++) {
            result += Integer.toString( ( hash[i] & 0xff ) + 0x100, 16).substring( 1 );
         }
         System.out.println(result);
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

VB 6.0

```
Private Const BITS_TO_A_BYTE = 8
Private Const BYTES_TO_A_WORD = 4
Private Const BITS_TO_A_WORD = 32

Private m_lOnBits(30)
Private m_l2Power(30)

   m_lOnBits(0) = CLng(1)
   m_lOnBits(1) = CLng(3)
   m_lOnBits(2) = CLng(7)
   m_lOnBits(3) = CLng(15)
   m_lOnBits(4) = CLng(31)
   m_lOnBits(5) = CLng(63)
   m_lOnBits(6) = CLng(127)
   m_lOnBits(7) = CLng(255)
   m_lOnBits(8) = CLng(511)
   m_lOnBits(9) = CLng(1023)
   m_lOnBits(10) = CLng(2047)
   m_lOnBits(11) = CLng(4095)
   m_lOnBits(12) = CLng(8191)
   m_lOnBits(13) = CLng(16383)
   m_lOnBits(14) = CLng(32767)
   m_lOnBits(15) = CLng(65535)
   m_lOnBits(16) = CLng(131071)
   m_lOnBits(17) = CLng(262143)
   m_lOnBits(18) = CLng(524287)
   m_lOnBits(19) = CLng(1048575)
   m_lOnBits(20) = CLng(2097151)
   m_lOnBits(21) = CLng(4194303)
   m_lOnBits(22) = CLng(8388607)
   m_lOnBits(23) = CLng(16777215)
   m_lOnBits(24) = CLng(33554431)
   m_lOnBits(25) = CLng(67108863)
   m_lOnBits(26) = CLng(134217727)
   m_lOnBits(27) = CLng(268435455)
   m_lOnBits(28) = CLng(536870911)
   m_lOnBits(29) = CLng(1073741823)
   m_lOnBits(30) = CLng(2147483647)

   m_l2Power(0) = CLng(1)
   m_l2Power(1) = CLng(2)
   m_l2Power(2) = CLng(4)
   m_l2Power(3) = CLng(8)
   m_l2Power(4) = CLng(16)
   m_l2Power(5) = CLng(32)
```

```
    m_l2Power(6) = CLng(64)
    m_l2Power(7) = CLng(128)
    m_l2Power(8) = CLng(256)
    m_l2Power(9) = CLng(512)
    m_l2Power(10) = CLng(1024)
    m_l2Power(11) = CLng(2048)
    m_l2Power(12) = CLng(4096)
    m_l2Power(13) = CLng(8192)
    m_l2Power(14) = CLng(16384)
    m_l2Power(15) = CLng(32768)
    m_l2Power(16) = CLng(65536)
    m_l2Power(17) = CLng(131072)
    m_l2Power(18) = CLng(262144)
    m_l2Power(19) = CLng(524288)
    m_l2Power(20) = CLng(1048576)
    m_l2Power(21) = CLng(2097152)
    m_l2Power(22) = CLng(4194304)
    m_l2Power(23) = CLng(8388608)
    m_l2Power(24) = CLng(16777216)
    m_l2Power(25) = CLng(33554432)
    m_l2Power(26) = CLng(67108864)
    m_l2Power(27) = CLng(134217728)
    m_l2Power(28) = CLng(268435456)
    m_l2Power(29) = CLng(536870912)
    m_l2Power(30) = CLng(1073741824)

Private Function LShift(lValue, iShiftBits)
    If iShiftBits = 0 Then
        LShift = lValue
        Exit Function
    ElseIf iShiftBits = 31 Then
        If lValue And 1 Then
            LShift = &H80000000
        Else
            LShift = 0
        End If
        Exit Function
    ElseIf iShiftBits < 0 Or iShiftBits > 31 Then
        Err.Raise 6
    End If

    If (lValue And m_l2Power(31 - iShiftBits)) Then
        LShift = ((lValue And m_lOnBits(31 - (iShiftBits + 1))) * m_l2Power(iShiftBits)) Or &H80000000
    Else
        LShift = ((lValue And m_lOnBits(31 - iShiftBits)) * m_l2Power(iShiftBits))
    End If
End Function
```

```
Private Function RShift(lValue, iShiftBits)
   If iShiftBits = 0 Then
      RShift = lValue
      Exit Function
   ElseIf iShiftBits = 31 Then
      If lValue And &H80000000 Then
         RShift = 1
      Else
         RShift = 0
      End If
      Exit Function
   ElseIf iShiftBits < 0 Or iShiftBits > 31 Then
      Err.Raise 6
   End If

   RShift = (lValue And &H7FFFFFFE) \ m_l2Power(iShiftBits)

   If (lValue And &H80000000) Then
      RShift = (RShift Or (&H40000000 \ m_l2Power(iShiftBits - 1)))
   End If
End Function

Private Function RotateLeft(lValue, iShiftBits)
   RotateLeft = LShift(lValue, iShiftBits) Or RShift(lValue, (32 - iShiftBits))
End Function

Private Function AddUnsigned(lX, lY)
   Dim lX4
   Dim lY4
   Dim lX8
   Dim lY8
   Dim lResult

   lX8 = lX And &H80000000
   lY8 = lY And &H80000000
   lX4 = lX And &H40000000
   lY4 = lY And &H40000000

   lResult = (lX And &H3FFFFFFF) + (lY And &H3FFFFFFF)

   If lX4 And lY4 Then
      lResult = lResult Xor &H80000000 Xor lX8 Xor lY8
   ElseIf lX4 Or lY4 Then
      If lResult And &H40000000 Then
         lResult = lResult Xor &HC0000000 Xor lX8 Xor lY8
      Else
         lResult = lResult Xor &H40000000 Xor lX8 Xor lY8
      End If
```

```
    Else
        lResult = lResult Xor lX8 Xor lY8
    End If

    AddUnsigned = lResult
End Function

Private Function F(x, y, z)
    F = (x And y) Or ((Not x) And z)
End Function

Private Function G(x, y, z)
    G = (x And z) Or (y And (Not z))
End Function

Private Function H(x, y, z)
    H = (x Xor y Xor z)
End Function

Private Function I(x, y, z)
    I = (y Xor (x Or (Not z)))
End Function

Private Sub FF(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(F(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub

Private Sub GG(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(G(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub

Private Sub HH(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(H(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub

Private Sub II(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(I(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub

Private Function ConvertToWordArray(sMessage)
```

```
    Dim lMessageLength
    Dim lNumberOfWords
    Dim lWordArray()
    Dim lBytePosition
    Dim lByteCount
    Dim lWordCount

    Const MODULUS_BITS = 512
    Const CONGRUENT_BITS = 448

    lMessageLength = Len(sMessage)

    lNumberOfWords = (((lMessageLength + ((MODULUS_BITS - CONGRUENT_BITS) \ BITS_TO_A_BYTE)) \
(MODULUS_BITS \ BITS_TO_A_BYTE)) + 1) * (MODULUS_BITS \ BITS_TO_A_WORD)
    ReDim lWordArray(lNumberOfWords - 1)

    lBytePosition = 0
    lByteCount = 0
    Do Until lByteCount >= lMessageLength
        lWordCount = lByteCount \ BYTES_TO_A_WORD
        lBytePosition = (lByteCount Mod BYTES_TO_A_WORD) * BITS_TO_A_BYTE
        lWordArray(lWordCount) = lWordArray(lWordCount) Or LShift(Asc(Mid(sMessage, lByteCount + 1,
1)), lBytePosition)
        lByteCount = lByteCount + 1
    Loop

    lWordCount = lByteCount \ BYTES_TO_A_WORD
    lBytePosition = (lByteCount Mod BYTES_TO_A_WORD) * BITS_TO_A_BYTE

    lWordArray(lWordCount) = lWordArray(lWordCount) Or LShift(&H80, lBytePosition)

    lWordArray(lNumberOfWords - 2) = LShift(lMessageLength, 3)
    lWordArray(lNumberOfWords - 1) = RShift(lMessageLength, 29)

    ConvertToWordArray = lWordArray
End Function

Private Function WordToHex(lValue)
    Dim lByte
    Dim lCount

    For lCount = 0 To 3
        lByte = RShift(lValue, lCount * BITS_TO_A_BYTE) And m_lOnBits(BITS_TO_A_BYTE - 1)
        WordToHex = WordToHex & Right("0" & Hex(lByte), 2)
    Next
End Function

Public Function MD5(sMessage)
```

```
Dim x
Dim k
Dim AA
Dim BB
Dim CC
Dim DD
Dim a
Dim b
Dim c
Dim d

Const S11 = 7
Const S12 = 12
Const S13 = 17
Const S14 = 22
Const S21 = 5
Const S22 = 9
Const S23 = 14
Const S24 = 20
Const S31 = 4
Const S32 = 11
Const S33 = 16
Const S34 = 23
Const S41 = 6
Const S42 = 10
Const S43 = 15
Const S44 = 21

x = ConvertToWordArray(sMessage)

a = &H67452301
b = &HEFCDAB89
c = &H98BADCFE
d = &H10325476

For k = 0 To UBound(x) Step 16
   AA = a
   BB = b
   CC = c
   DD = d

   FF a, b, c, d, x(k + 0), S11, &HD76AA478
   FF d, a, b, c, x(k + 1), S12, &HE8C7B756
   FF c, d, a, b, x(k + 2), S13, &H242070DB
   FF b, c, d, a, x(k + 3), S14, &HC1BDCEEE
   FF a, b, c, d, x(k + 4), S11, &HF57C0FAF
   FF d, a, b, c, x(k + 5), S12, &H4787C62A
   FF c, d, a, b, x(k + 6), S13, &HA8304613
```

FF b, c, d, a, x(k + 7), S14, &HFD469501
FF a, b, c, d, x(k + 8), S11, &H698098D8
FF d, a, b, c, x(k + 9), S12, &H8B44F7AF
FF c, d, a, b, x(k + 10), S13, &HFFFF5BB1
FF b, c, d, a, x(k + 11), S14, &H895CD7BE
FF a, b, c, d, x(k + 12), S11, &H6B901122
FF d, a, b, c, x(k + 13), S12, &HFD987193
FF c, d, a, b, x(k + 14), S13, &HA679438E
FF b, c, d, a, x(k + 15), S14, &H49B40821

GG a, b, c, d, x(k + 1), S21, &HF61E2562
GG d, a, b, c, x(k + 6), S22, &HC040B340
GG c, d, a, b, x(k + 11), S23, &H265E5A51
GG b, c, d, a, x(k + 0), S24, &HE9B6C7AA
GG a, b, c, d, x(k + 5), S21, &HD62F105D
GG d, a, b, c, x(k + 10), S22, &H2441453
GG c, d, a, b, x(k + 15), S23, &HD8A1E681
GG b, c, d, a, x(k + 4), S24, &HE7D3FBC8
GG a, b, c, d, x(k + 9), S21, &H21E1CDE6
GG d, a, b, c, x(k + 14), S22, &HC33707D6
GG c, d, a, b, x(k + 3), S23, &HF4D50D87
GG b, c, d, a, x(k + 8), S24, &H455A14ED
GG a, b, c, d, x(k + 13), S21, &HA9E3E905
GG d, a, b, c, x(k + 2), S22, &HFCEFA3F8
GG c, d, a, b, x(k + 7), S23, &H676F02D9
GG b, c, d, a, x(k + 12), S24, &H8D2A4C8A

HH a, b, c, d, x(k + 5), S31, &HFFFA3942
HH d, a, b, c, x(k + 8), S32, &H8771F681
HH c, d, a, b, x(k + 11), S33, &H6D9D6122
HH b, c, d, a, x(k + 14), S34, &HFDE5380C
HH a, b, c, d, x(k + 1), S31, &HA4BEEA44
HH d, a, b, c, x(k + 4), S32, &H4BDECFA9
HH c, d, a, b, x(k + 7), S33, &HF6BB4B60
HH b, c, d, a, x(k + 10), S34, &HBEBFBC70
HH a, b, c, d, x(k + 13), S31, &H289B7EC6
HH d, a, b, c, x(k + 0), S32, &HEAA127FA
HH c, d, a, b, x(k + 3), S33, &HD4EF3085
HH b, c, d, a, x(k + 6), S34, &H4881D05
HH a, b, c, d, x(k + 9), S31, &HD9D4D039
HH d, a, b, c, x(k + 12), S32, &HE6DB99E5
HH c, d, a, b, x(k + 15), S33, &H1FA27CF8
HH b, c, d, a, x(k + 2), S34, &HC4AC5665

II a, b, c, d, x(k + 0), S41, &HF4292244
II d, a, b, c, x(k + 7), S42, &H432AFF97
II c, d, a, b, x(k + 14), S43, &HAB9423A7
II b, c, d, a, x(k + 5), S44, &HFC93A039

```
        II a, b, c, d, x(k + 12), S41, &H655B59C3
        II d, a, b, c, x(k + 3), S42, &H8F0CCC92
        II c, d, a, b, x(k + 10), S43, &HFFEFF47D
        II b, c, d, a, x(k + 1), S44, &H85845DD1
        II a, b, c, d, x(k + 8), S41, &H6FA87E4F
        II d, a, b, c, x(k + 15), S42, &HFE2CE6E0
        II c, d, a, b, x(k + 6), S43, &HA3014314
        II b, c, d, a, x(k + 13), S44, &H4E0811A1
        II a, b, c, d, x(k + 4), S41, &HF7537E82
        II d, a, b, c, x(k + 11), S42, &HBD3AF235
        II c, d, a, b, x(k + 2), S43, &H2AD7D2BB
        II b, c, d, a, x(k + 9), S44, &HEB86D391

        a = AddUnsigned(a, AA)
        b = AddUnsigned(b, BB)
        c = AddUnsigned(c, CC)
        d = AddUnsigned(d, DD)
    Next

    MD5 = UCase(WordToHex(a) & WordToHex(b) & WordToHex(c) & WordToHex(d))
End Function
```

Annex 4 – Convert from UCS2 to Arabic (or UTF8) function

VB Script and VB.net :

```
Function Arabize(ByVal text As String) As String
      Try
          Dim thetext, char1, arabictext As String
          arabictext = ""
          thetext = text
          For i As Integer = 1 To Len(thetext) / 4
              char1 = Left(thetext, 4)
              arabictext = arabictext & ChrW(CInt("&H" & char1))
              thetext = Right(thetext, Len(thetext) - 4)
          Next
          Arabize = arabictext
      Catch ex As Exception
          Arabize = " invalid message"
      End Try
  End Function
```

PHP 5

```
header('Content-Type: text/html; charset=UTF-8');
mb_http_output('UTF-8');
echo '<html><head>';
echo '<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />';
echo '</head><body>';

$UCS2 = "06230643062A064A064400200647064A002006270644062306420648064 9";
$ucs2string = pack("H*" , $_REQUEST['ucs2']);
//$ucs2string = pack("H*" , $UCS2);
$utf8string = mb_convert_encoding($ucs2string , 'UTF-8', 'UCS-2');
echo 'UTF8: '.$utf8string.'<br />';

echo '</body></html>';
```

Annex 5 – Convert from Arabic (or UTF8) to UCS2 function

VB Script and VB.net:

```
  Public Function UniEnc(ByVal text As String) As String

        Dim i, j, count_char As Integer, nchar, hexval, arabicunicode As
String
        arabicunicode = ""
        For i = 1 To Len(text)
            count_char = count_char + 1
            nchar = Left(text, 1)
            hexval = Hex(AscW(nchar))
            If Len(hexval) < 4 Then
                For j = 1 To 4 - Len(hexval)
                    hexval = "0" & hexval
                Next
            End If
            arabicunicode = arabicunicode & hexval
            text = Right(text, Len(text) - 1)
        Next
        UniEnc = arabicunicode
    End Function
```

PHP 5

```
header('Content-Type: text/html; charset=UTF-8');
mb_http_output('UTF-8');
echo '<html><head>';
echo '<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />';
echo '</head><body>';

$bodyar = "الأقوى هي أكتيل!";
$arr = unpack('H*hex', @iconv('UTF-8', 'UCS-2BE', $bodyar));
echo "UCS2 : ".$arr['hex'];
echo "<br><br>";

echo '</body></html>';
```