Steps to enable SSL Client/Server for Microchip TCP/IP Stack v5.36.04.

These changes are required to support SSL when using TCP/IP Stack Version 5.36.04 and Microchip's Data Encryption Libraries v2.6 (or earlier). Start with the October, 2011 Microchip Applications Libraries release (containing TCP/IP Stack Version 5.36.04) and the Microchip Data Encryption Libraries for the TCP/IP Stack (SW300052 v2.6). Make the following changes to the indicated file(s):

[MAL Installation Directory]/Microchip/TCPIP Stack/RSA.c (this file is installed by the Data Encryption Libraries installer)

1. change

   ```
   BOOL RSABeginUsage(RSA_OP op, BYTE vKeyByteLen)
   ```

   to

   ```
   BOOL RSABeginUsage(RSA_OP op, WORD vKeyByteLen)
   ```

2. change

   ```
   void RSASetData(BYTE* data, BYTE len, RSA_DATA_FORMAT format)
   ```

   to

   ```
   void RSASetData(BYTE* data, WORD len, RSA_DATA_FORMAT format)
   ```

3. change

   ```
   static BYTE keyLength;
   ```

   to

   ```
   static WORD keyLength;
   ```

4. Remove this from inside the conditional #if defined(STACK_USE_RSA_ENCRYPT) on line 81 :

   ```
   #if defined(__18CXX) && !defined(HI_TECH_C)
           #pragma udata RSA_TEMP_SPACE
   #endif

   BYTE rsaTemp[256];      // Temporary data storage space for encryption
   ```

5. Add the following code section at the bottom of the “Global RSA Variables” section (outside of the “#if defined (STACK_USE_RSA_DECRYPT)” block).

```
#if defined(__18CXX) && !defined(HI_TECH_C)
        #pragma udata RSA_TEMP_SPACE
#endif

BYTE rsaTemp[SSL_RSA_CLIENT_SIZE/4]; // Temporary data storage space for encryption/decryption

#if defined(__18CXX) && !defined(HI_TECH_C)
        #pragma udata
#endif
```

6. change

```
BYTE rsaData[128];
```

to

```
BYTE rsaData[SSL_RSA_CLIENT_SIZE / 8];
```

7. Move the following line from inside the #if defined (STACK_USE_RSA_ENCRYPT) section of RSAInit(void) :

```
BigInt(&tmp, (BIGINT_DATA_TYPE*)rsaTemp, sizeof(rsaTemp)/sizeof(BIGINT_DATA_TYPE));
```

To outside of the #if #else #endif block

8. Remove the following line from the corresponding #else block (and remove the #else preprocessor directive) :

```
BigInt(&tmp, (BIGINT_DATA_TYPE*)&sslBuffer.full[128], 128/sizeof(BIGINT_DATA_TYPE));
```

9. In the function RSAStep(void) :

Change

```
BigInt(&m1, (BIGINT_DATA_TYPE*)((BYTE*)&sslBuffer+64), RSA_PRIME_WORDS);
```

to

```
BigInt(&m1, (BIGINT_DATA_TYPE*)((BYTE*)&sslBuffer+(SSL_RSA_KEY_SIZE/8)), RSA_PRIME_WORDS);
```

and change

```
BigInt(&m2, (BIGINT_DATA_TYPE*)((BYTE*)&sslBuffer+96), RSA_PRIME_WORDS);
```

to

```
BigInt(&m2, (BIGINT_DATA_TYPE*)((BYTE*)&sslBuffer+(3*(SSL_RSA_KEY_SIZE/16))), RSA_PRIME_WORDS);
```