To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

Renesas MCU

Perl Script to Convert S-Record Data to a C Structure

Introduction

Application note REG05B0021-0100 – Utility to Convert S-Record Data to a C Structure – provides and describes a Win32 command line utility, ‘motice_cl’, to perform the described conversion. Such a utility is useful when writing flash programming routines when code must be executed from RAM but stored in flash memory until required. Although ‘motice_cl’ performs the conversion process well there are situations when it is desirable to have the output file formatted differently or when software is being developed on a different operating system such as Linux. In such situations it is either not easy or possible to use ‘motice_cl’.

For these reasons a perl (portable extraction and report language) script has been produced to perform the same function as ‘motice_cl’. However, as the script by its very nature is provided as a source file its functionality can easily be modified if different output formatting etc is desired. In addition so long as there is an implementation of perl available the script can be run on a multitude of operating systems.

The script is included in the appendix of this apps note and has been tested on WindowsXP® using ActivePerl by ActiveState (http://www.activestate.com) and under Cygwin (http://www.cygwin.com) and also on Linux. Of course the output of this script should be fully tested before being used in a production environment.

As with ‘motice_cl’ calling of this perl script can be integrated into HEW as a build phase when developing on a Windows® platform.

It is expected that the reader has a basic understanding of perl when reading this application note. Further information on perl can be obtained from the links above, Googling perl or visiting http://www.perl.org.

Contents

Using the Script

It is expected that apps note REG05B0021-0100 is read first before using the perl script provided here. This apps note describes why a conversion from an s-record file to a C structure might be useful and also how to integrate a command line utility into HEW as a build phase.

The text of the perl script can be copied from appendix A of this apps note and pasted into a file which will be called 'motice.pl' for the purpose of this apps note. The script is called using the command line shown below.

```
perl motice.pl source destination structname
```

Source:	Specifies the name and path of the s-record file to be processed.
Destination:	Specifies the name and path of the C file to be produced.
Structname:	Specifies the name of the C structure to be generated.

The C file output by 'motice.pl' contains the following structure definition.

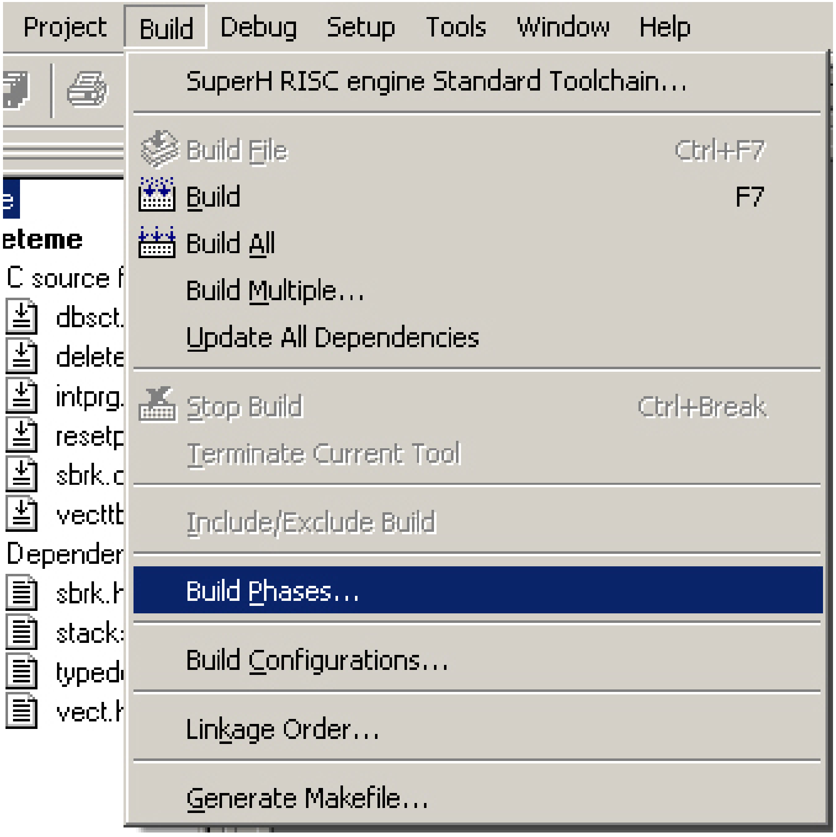
```
struct rom_data {
        unsigned long start_address;
        unsigned long data_length;
        unsigned char data[4708];
};
```

The name of the structure is dependent on the 'structname' command line parameter and the size of the 'data' array is dependent on the amount of data in the processed s-record. Further information can be found in apps note REG05B0021-0100.

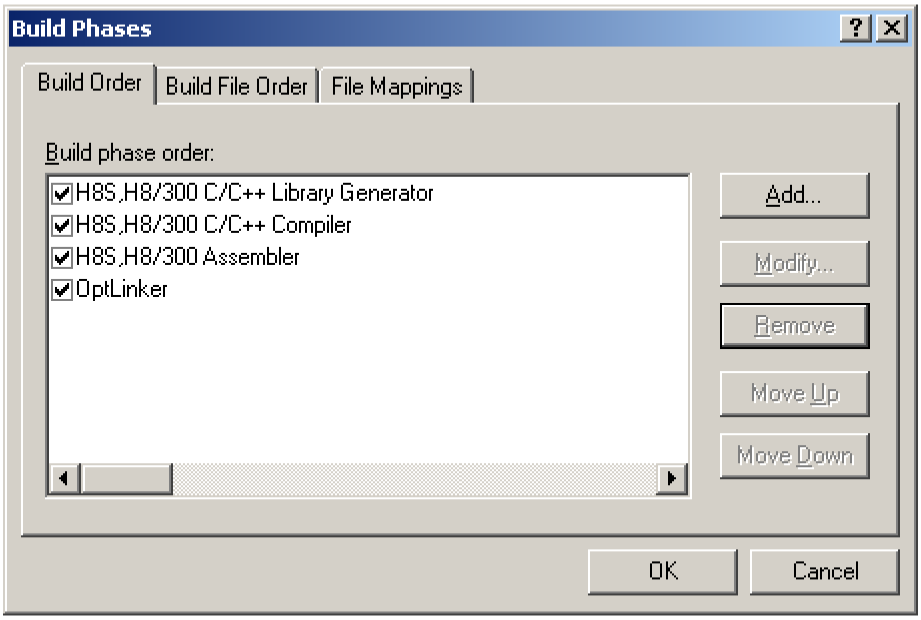
HEW Integration

HEW (High-Performance Embedded Workshop) is the Renesas IDE integrating the compiler toolchains and debuggers in a single application. One of the features of HEW is the ability to produce custom build phases. This lends itself well to this script as a separate build phase can run to call the utility each time an application is built. The following screenshots show the process of setting up the build phase.

1. Select 'Build Phases' from the 'Options' menu.



2. Click the add button.



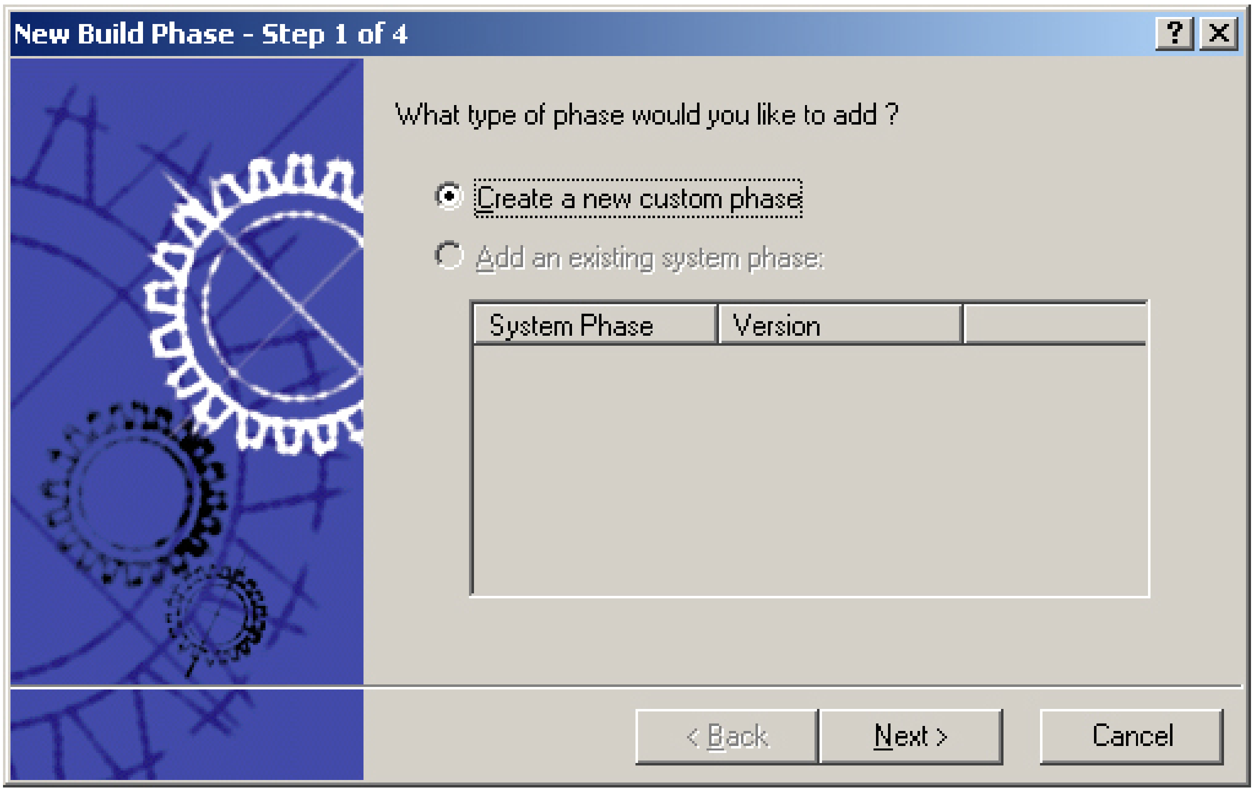
Build Phases

Build Order | Build File Order | File Mappings

Build phase order:

☑ H8S,H8/300 C/C++ Library Generator
☑ H8S,H8/300 C/C++ Compiler
☑ H8S,H8/300 Assembler
☑ OptLinker

3. Create a custom build phase by clicking next.



New Build Phase - Step 1 of 4

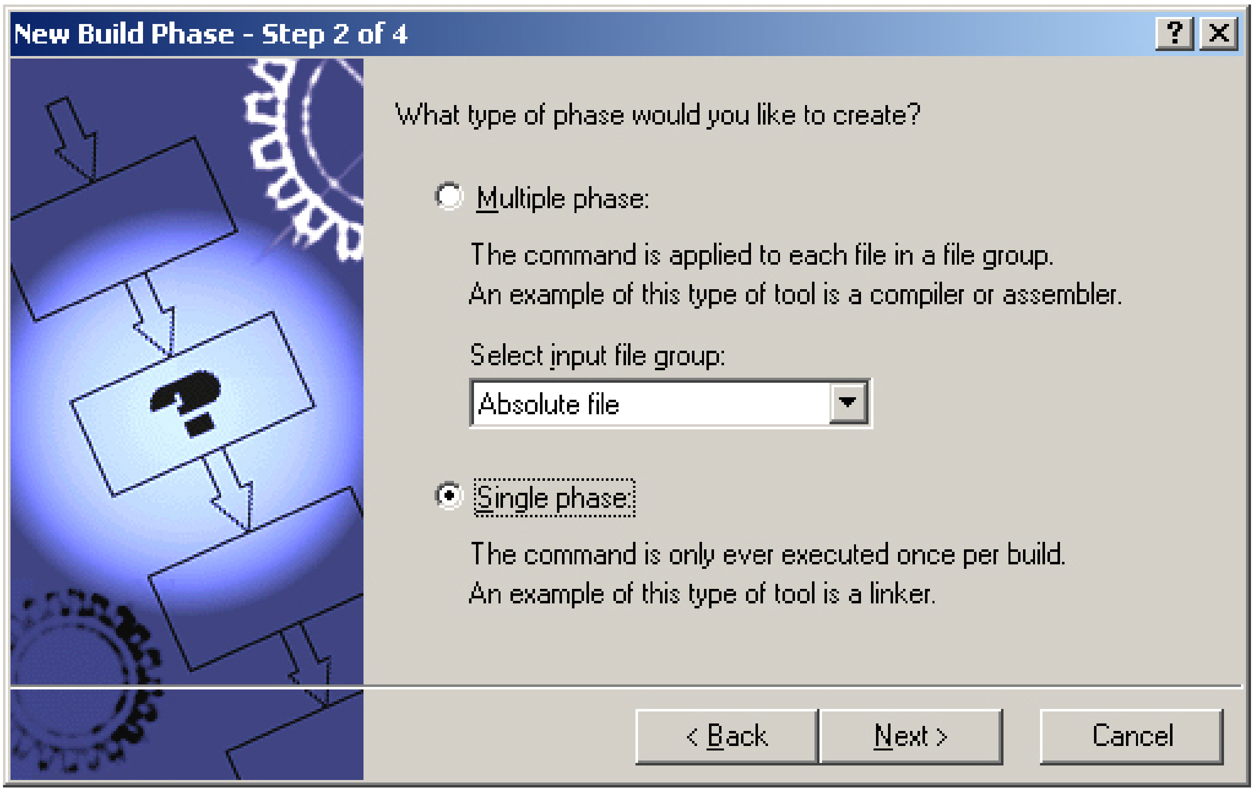
What type of phase would you like to add ?

● Create a new custom phase

○ Add an existing system phase:

System Phase	Version	

4. Create a single build phase. This means that the phase will only execute once per build. The 'motice.pl' script is only required to run once when it processes the s-record file output from the linker.



What type of phase would you like to create?

○ Multiple phase:

The command is applied to each file in a file group.
An example of this type of tool is a compiler or assembler.

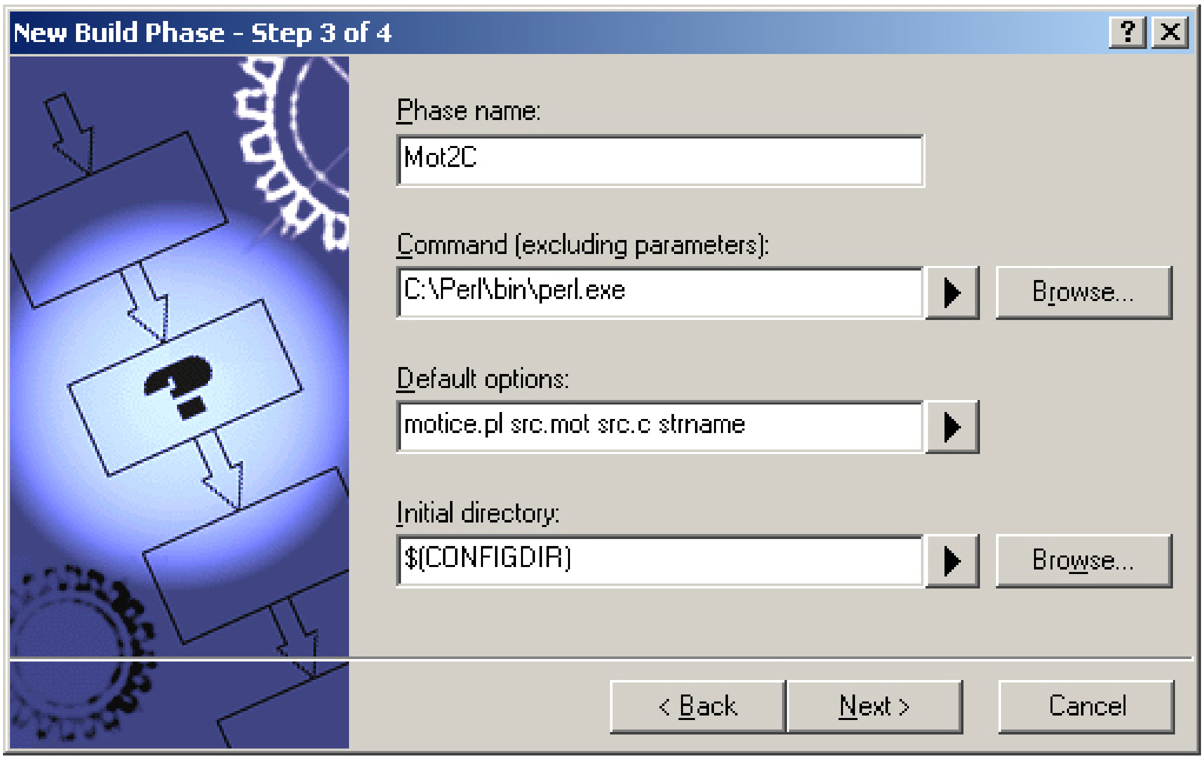
Select input file group: Absolute file

● Single phase:

The command is only ever executed once per build.
An example of this type of tool is a linker.

5. Configure the build phase to call the perl interpreter with the correct command line parameters

Here the phase has been called ‘Mot2c’, the full path for the perl interpreter executable has been entered as have the default parameters for the command including the script itself. The source file will be the s-record output from the HEW project and will usually be found in the debug or release build directories.



New Build Phase - Step 3 of 4

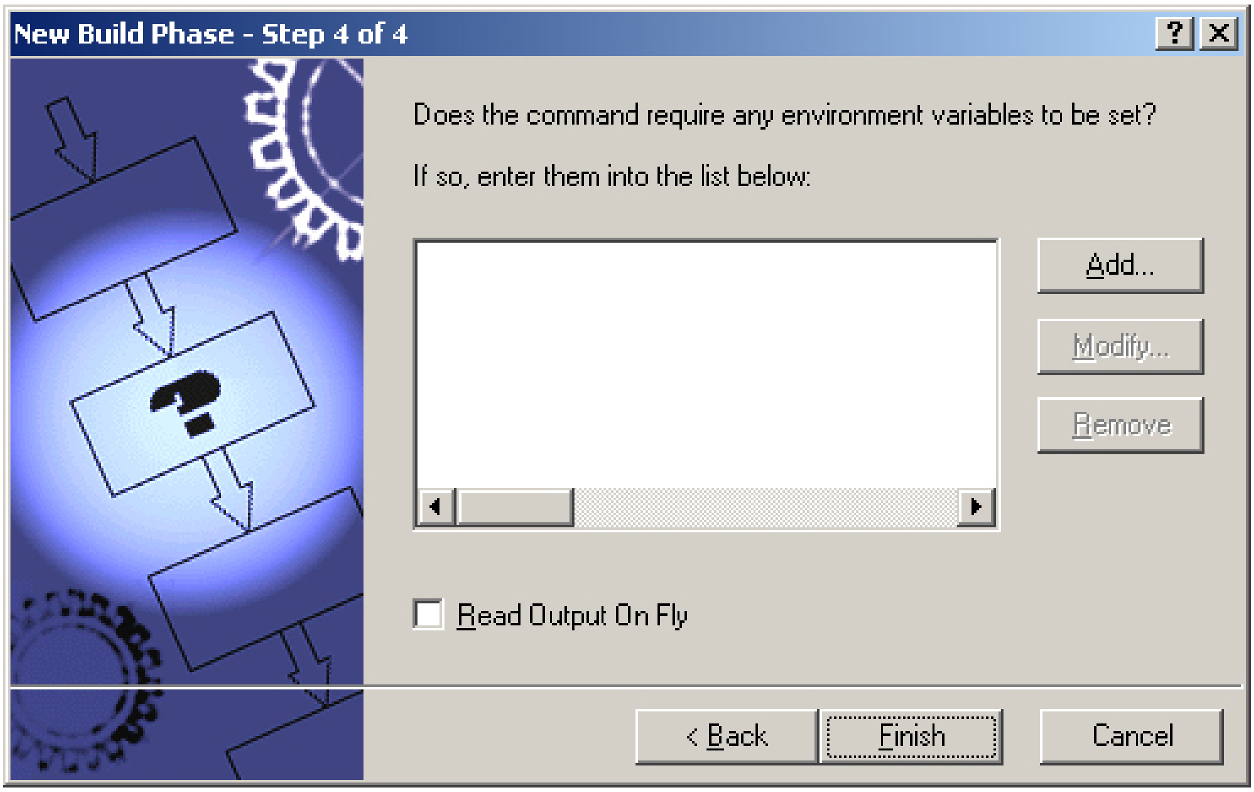
Phase name: Mot2C

Command (excluding parameters): C:\Perl\bin\perl.exe

Default options: motice.pl src.mot src.c strname

Initial directory: $(CONFIGDIR)

6. No environment variables are required for the utility so leave this blank and hit ‘Finish’.



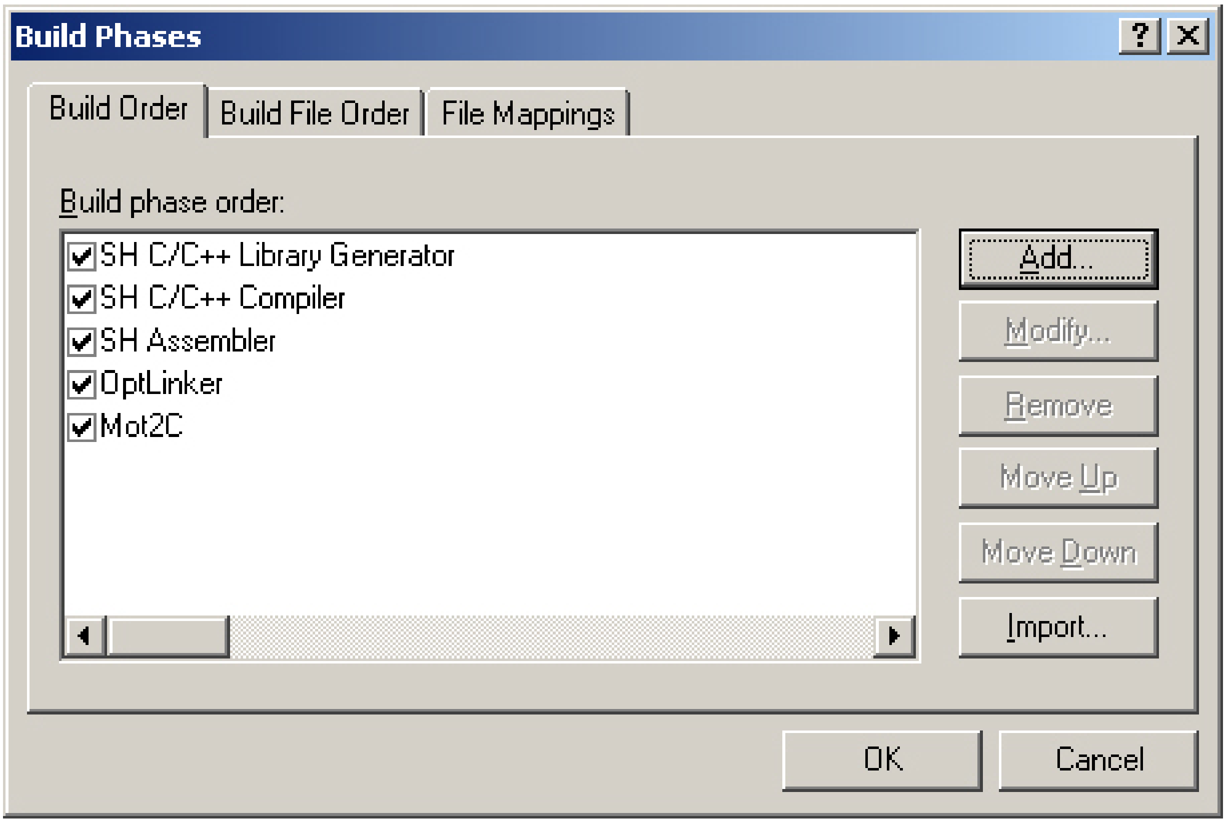
New Build Phase - Step 4 of 4

Does the command require any environment variables to be set?

If so, enter them into the list below:

□ Read Output On Fly

7. The ‘Build Phases’ form should now show the newly created ‘Mot2c’ build phase. Ensure that it is ticked so that the phase is executed. The order the build phases are executed can be changed. This is not necessary here as the ‘Mot2c’ phase must run after the linker otherwise it will have no s-record file to process!



☑SH C/C++ Library Generator
☑SH C/C++ Compiler
☑SH Assembler
☑OptLinker
☑Mot2C

Click ‘OK’. Now when the project is built the ‘Mot2c’ phase will execute and convert the s-record file to a C structure. The HEW output window will show the output from the script.

Summary

The script provided with this apps note provides the same functionality as the utility ‘motice_cl’ described by apps note REG05B0021-0100. As the script is written in the perl language and supplied as source code it is possible for it to be run on different platforms and for its functionality to be modified as required. The script is fully commented and so should be easy to understand and change.

Appendix A – Perl Script Source

```
#!/usr/bin/perl -w

# perl script to convert the data in a file of s-records into
# a constant C structure

# command line format for the script is
# motice.pl sourcefile destinationfile structname

print( "\nMotice.pl - v0.1 (c) Renesas 2005\n" );
print(   "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n" );

# check we have 3 command line arguments
if( ( $#ARGV + 1 ) != 3 )
{
 # incorrect number of command line arguments, print usage instructions
 print( "\nMotice.pl is a utility which converts the data within a
Motorola\n" );
 print( "S Record file into a C source file with the data in a C
structure.\n" );
 print( "\nperl motice.pl source destination structname\n" );
 print( "\n\tsource\t\tspecifies the s record source file\n" );
 print( "\tdestination\tspecifies the C file to be generated\n" );
 print( "\tstructname\tspecifies the name of the generated C structure\n" );
 die;
}

$total_records = 0;
$address = 0;
$count = 0;
$checksum = 0;
$min_address = 0xFFFFFFFF;
$max_address = 0;
$databytecount = 0;

# attempt to open source file - s-record
open( INFILE, $ARGV[0] ) || die "Cannot open source file: $ARGV[0]\n";
# read each s-record in turn to establish the maximum and minimum addresses of
data in the file
while( $srec = <INFILE> )
```

```
{
 chomp( $srec );
 $ret = process_srec();
 if ( $ret == 0 )
  {
  # pass
  if( $min_address > hex( $address ) )
  {
   $min_address = hex( $address );
  }

  if( $max_address < ( ( hex( $address ) + $databytecount - 1 ) ) )
  {
   $max_address = ( ( hex( $address ) + $databytecount - 1 ) );
   # '-1' is neaded here because if the address is 1000 and the byte count is 1
the
   # maximum address is 1000 (not 1001)
  }
 }
  else
  {
   # fail
   if( $ret == 1 )
   {
    # not an s-record
    close( INFILE );
    my $l = $total_records + 1;
    die( "\nError: Line $l is not a valid s-record\nExiting\n" );
   }
   elsif( $ret == 2 )
   {
    # checksum failure
    close( INFILE );
    die( "\nError: Line $total_records checksum error\nExiting\n" );
   }
   elsif( $ret == 3 )
   {
    # S0, S5, S7, S8 or S9 record
   }
  }
}
```

```
close( INFILE );

# create array with enough space to hold the data
# do this by placing an element at the end of the array
$alldata[ $max_address - $min_address ] = ();
# initialise contents to H'FF
$index=0;
foreach (@alldata)
{
 $alldata[ $index ] = 0xff;
 $index++;
}

# reopen s-record file and copy data contents into the array
open( INFILE, $ARGV[0] ) || die "Cannot open source file: $ARGV[0]\n";
$total_records = 0;
while( $srec = <INFILE> )
{
 chomp( $srec );
 $ret = process_srec();
 if ( $ret == 0 )
 {
  # pass
  # copy data into the array
  $index = 0;
  $srecstr = pack( "H*", $srecdata );           # convert hex values to binary
string
  @srecvals = unpack( "C*", $srecstr );      # convert binary values to unsigned
char bytes
  foreach $sval (@srecvals)
  {
   $alldata[ (hex($address) + $index ) - ($min_address) ] = $sval;
   $index++;
  }
 }
  else
  {
   # fail
   if( $ret == 1 )
   {
    # not an s-record
    # this should not happen as it should have been detected in the first pass
through the file
    close( INFILE );
    my $l = $total_records + 1;
    die( "\nError: Line $l is not a valid s-record\nExiting\n" );
   }
   elsif( $ret == 2 )
   {
    # checksum failure
    # this should not happen as it should have been detected in the first pass
through the file
    close( INFILE );
    die( "\nError: Line $total_records checksum error\nExiting\n" );
   }
   elsif( $ret == 3 )
   {
    # S0, S5, S7, S8 or S9 record
   }
  }
}
close( INFILE );

################
# Write C File #
################
# open file to write to
open( OUTFILE, ">$ARGV[1]" ) || die "Cannot open destination file: $ARGV[1]\n";

print OUTFILE ( "//\n" );
print OUTFILE ( "//\tThis file has been generated by 'motice.pl' (v0.1) (c)
Renesas 2005\n" );
print OUTFILE ( "//\n" );
print OUTFILE ( "//\t$ARGV[1]\n" );
my $tad = localtime;                    # get the system date and time
print OUTFILE ( "//\t$tad\n" );
print OUTFILE ( "\n\n" );

print OUTFILE ( "struct rom_data {\n" );
print OUTFILE ( "\tunsigned long start_address;\n");
print OUTFILE ( "\tunsigned long data_length;\n");
print OUTFILE ( sprintf( "\tunsigned char data[%d];\n", ($max_address -
$min_address) + 1 ) );
```

```
# add 1 to max - min address as if min was 0 and max was 2 there would be 3 (0,
1 and 2 ) data bytes
print OUTFILE ( "};\n" );


print OUTFILE ( "\nconst struct rom_data $ARGV[2] = {\n" );
print OUTFILE ( sprintf( "\t0x%08x,\n", $min_address ) );
print OUTFILE ( sprintf( "\t0x%08x,", ($max_address - $min_address) + 1 ) );
# add 1 to max - min address as if min was 0 and max was 2 there would be 3 (0,
1 and 2 ) data bytes


$index = 0;
foreach $sval (@alldata)
{
 if( $index % 16 == 0 )  # 16 bytes of data per line
 {
  print OUTFILE ( "\n\t" ) ;
 }
 print OUTFILE ( sprintf( "0x%02x", $sval ) );
 # only add a comma if the end of the C array has not been reached
 if( $index != ( $max_address - $min_address ) ) { print OUTFILE ( ", " ); }
 $index++;
}


print OUTFILE ( "\n};\n" );
close( OUTFILE );


# print stats to console
print( "\nSource file\t= $ARGV[0]\n" );
print( "Dest file\t= $ARGV[1]\n" );
print( "Struct name\t= $ARGV[2]\n" );
print( sprintf("Start address\t= 0x%x\n",$min_address) );
print( sprintf("End address\t= 0x%x\n",$max_address) );
print( sprintf( "Byte count\t= 0x%x (%d bytes)\n", ($max_address - $min_address)
+ 1, ($max_address - $min_address) + 1 ) );
# add 1 to max - min address as if min was 0 and max was 2 there would be 3 (0,
1 and 2 ) data bytes
print( "$total_records s-record(s) processed" );
#end


################
# sub-routines #
################
```

```
sub process_srec {
  if( $srec =~ /^s[0-9]/i )                     # does string start with "S" or
"s" and a digit 0 to 9
  {
   # looks like an s-record
   # extract type
   $type = hex( substr( $srec, 1, 1 ) );
  }
  else
  {
   # not an s-record record
   return 1;
  }

  #extract length, address, checksum and data from record types S1, S2 and S3
  if( ( $type == 1 ) || ( $type == 2 ) || ( $type == 3 ) )
  {
   $total_records++;
   $count = substr( $srec, 2, 2 );
   $databytecount = ( ( hex( $count ) ) - ( $type + 1 ) - 1 );
   $address = substr( $srec, 4, ( ( $type + 1 ) * 2 ) );
   $checksum = substr( $srec, ( 4 + ( ( ( hex ($count) ) - 1 ) * 2 ) ), 2  );

   # verify the checksum
   # checksum is an 8-bit hex value representing the 1's complement of the sum
of byte values of
   # the count, address and data fields
   $cs = 0;
   for( $i=2; $i<( ( ( hex ( $count ) ) * 2 ) + 2 ); $i+=2 )
   {
    $cs += hex( substr( $srec, $i, 2 ) );
   }
   $cs = $cs % 256;
   $cs = 255 - $cs;

   # check the checksum
   if( $cs == hex( $checksum ) )
   {
    # checksum passed
    # extract the data
    $srecdata = substr( $srec, 4 + ( ( $type + 1 ) * 2 ), 2 * $databytecount );
   }
   else
   {
    # checksum fail
    return 2;
   }
  }
  elsif( ( $type == 0 ) || ( $type == 5 ) || ( $type == 7 ) || ( $type == 8 ) ||
( $type == 9 ) )
  {
   $total_records++;
   return 3;
  }
  else
  {
   # not an s-record - should not get here
   return 1;
  }

  return 0;
} #end of sub process_srec
```

Website and Support

Renesas Technology Website
  http://www.renesas.com/

Inquiries
  http://www.renesas.com/inquiry
  csc@renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.