FAQ 011

Adding Paths for S-Function Source Files

Keywords

additional paths; S-function source files; user makefile; USR.MK file, rtwmakecfg.m

Question

I have organized my S-function MEX DLL files, source files and header files in separate folders. When using an S-function in a model I would like to automatically add the respective source paths to the build process. How can I do that?

Relevant Products

Real-Time Interface (RTI) and ConfigurationDesk, differences between Real-Time Interface (RTI) and ConfigurationDesk are explicitly pointed to

Solution

You can choose from two methods to automatically add paths for S-function source and header files to the build process:

- To add model-specific paths
- To add paths for all models

The second method is a feature of the Simulink Coder® (formerly Real-Time Workshop®).

To add model-specific paths

- **For RTI**: Edit the model's user makefile template `<modelname>_usr.mk`. Enter the necessary folders at the `SFCN_DIR` and `USER_INCLUDES_PATH` macros.
- **For ConfigurationDesk and RTI**: Edit the Configuration Parameters' custom code page settings beneath the Code Generation page (formerly Real-Time Workshop page with Real-Time Workshop). Enter the necessary folders (space separated list) at "Include list of additional…Include directories" (valid for RTI and ConfigurationDesk) (*FAQ 271*).

To add paths for all models (valid for RTI and ConfigurationDesk)

1. Create a `rtwmakecfg.m` file in the folder that holds MEX DLL files. If a model contains an S-function with the MEX DLL file in this folder, the `rtwmakecfg.m` file provides the build process with the information about what folders contain the S-function source and header files.

2. Open the `rtwmakecfg.m` file in an editor and create an `rtwmakecfg()` function, which returns a struct with an `includePath` field and a `sourcePath` field. The example shown below can be used as a template.

   For further information, refer to *Using the rtwmakecfg.m API to Customize Generated Makefiles* in the *Simulink Coder User's Guide* by *The MathWorks.*

Assume that you have the following directory structure:

```
C:\Projects\SFunctions\
rtwmakecfg.m
sfun1.mexw32 (up to MATLAB® R14SP2 sfun1.dll)
sfun2.mexw32 (up to MATLAB® R14SP2 sfun2.dll)

.\Sources\
sfun1.c
sfun2.c

.\Headers\
sfun1.h
sfun2.h
sfun_common.h
```

Then the related `rtwmakecfg.m` file looks like this:

```
function makeInfo = rtwmakecfg()
% RTWMAKECFG Add include and source directories to
% Simulink Coder make files.
% makeInfo = RTWMAKECFG returns a structured array containing
% following fields:

% makeInfo.includePath - cell array containing additional include
% directories. Those directories will be expanded into include
% instructions of Simulink Coder generated make files.

% makeInfo.sourcePath - cell array containing additional source
% directories. Those directories will be expanded into rules of
% Simulink Coder generated make files.

% makeInfo.library - structure containing additional runtime library
% names and module objects. This information will be expanded into
% rules of generated make files.
% ... .library(1).Name - name of runtime library
% ... .library(1).Modules - cell array containing source file names
% for the runtime library
% ... .library(2).Name
% ... .library(2).Modules

% This RTWMAKECFG file must be located in the same directory as the
% related S-function MEX-DLL(s). If one or more S-functions of the
% directory are referenced by a model Simulink Coder will evaluate
% RTWMAKECFG to obtain the additional include and source directories.

% To examine more RTWMAKECFG files in your installation issue at the
% MATLAB prompt:
% >> which RTWMAKECFG -all

% Issue a message.
seperatorLine = char(ones(1,70) * '~');
fprintf('\n');
fprintf('%s\n', seperatorLine);
fprintf(' %s\n', which(mfilename));
fprintf(' Adding source and include directories to make process.\n')

% Setting up the return structure with
% - source directories:
% C:\Projects\SFunctions\Sources
makeInfo.sourcePath = { ...
'C:\Projects\SFunctions\Sources' ...
};

% - include directories
% C:\Projects\SFunctions\Headers
makeInfo.includePath = { ...
'C:\Projects\SFunctions\Headers' ...
};

% Display contents.
fprintf(' - additional source directories:\n');
fprintf(' %s\n', makeInfo.sourcePath{:});
fprintf(' - additional include directories:\n');
fprintf(' %s\n', makeInfo.includePath{:});
fprintf('%s\n', seperatorLine);

% [EOF] rtwmakecfg.m
```

Related dSPACE HelpDesk documents

- *User Makefile (USR.MK File)* in the
  RTI and RTI-MP Implementation Reference.

- *Using custom code* in the
  ConfigurationDesk Simulink Modelling Guide

Related FAQ

- *FAQ 271:* Alternatives for the User-Code and User-Make file

How to Contact dSPACE Support

dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

++49 5251 1638-941

mailto:support@dspace.de
http://www.dspace.com/support

dSPACE recommends that you use the support request form on the Internet to contact dSPACE Support.
It is available at:

- http://www.dspace.com/go/supportrequest

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

FAQ

FAQ documents are available at http://www.dspace.com/go/faq.

Important Notice

This publication and the contents hereof are subject to change without notice.

A list of registered dSPACE trademarks is available at:
http://www.dspace.com/go/Trademarks