Feature names are case-sensitive and may only contain letters, numbers

and underscores. Feature names defined in all uppercase are reserved for

CMake's own built-in features (see `Predefined Features`\_ further below).

Feature Definitions

^^^^^^^^^^^^^^^^^^^

A group feature definition is a list that contains exactly two elements:

::

<PREFIX> <SUFFIX>

On the linker command line, ``<PREFIX>`` will precede the list of libraries

in the group and ``<SUFFIX>`` will follow after.

For the elements of this variable, the ``LINKER:`` prefix can be used.

.. include:: ../command/LINK\_OPTIONS\_LINKER.txt

:start-line: 3

Examples

^^^^^^^^

Solving cross-references between two static libraries

"""""""""""""""""""""""""""""""""""""""""""""""""""""

A project may define two or more static libraries which have circular

dependencies between them. In order for the linker to resolve all symbols

at link time, it may need to search repeatedly among the libraries until no

new undefined references are created. Different linkers use different syntax

for achieving this. The following example shows how this may be implemented

for some linkers. Note that this is for illustration purposes only.

Projects should use the built-in ``RESCAN`` group feature instead

(see `Predefined Features`\_), which provides a more complete and more robust

implementation of this functionality.

.. code-block:: cmake

set(CMAKE\_C\_LINK\_GROUP\_USING\_cross\_refs\_SUPPORTED TRUE)

if(CMAKE\_C\_COMPILER\_ID STREQUAL "GNU" AND CMAKE\_SYSTEM\_NAME STREQUAL "Linux")

set(CMAKE\_C\_LINK\_GROUP\_USING\_cross\_refs

"LINKER:--start-group"

"LINKER:--end-group"

)

elseif(CMAKE\_C\_COMPILER\_ID STREQUAL "SunPro" AND CMAKE\_SYSTEM\_NAME STREQUAL "SunOS")

set(CMAKE\_C\_LINK\_GROUP\_USING\_cross\_refs

"LINKER:-z,rescan-start"

"LINKER:-z,rescan-end"

)

else()

# feature not yet supported for the other environments

set(CMAKE\_C\_LINK\_GROUP\_USING\_cross\_refs\_SUPPORTED FALSE)

endif()

add\_library(lib1 STATIC ...)

add\_library(lib2 SHARED ...)

if(CMAKE\_C\_LINK\_GROUP\_USING\_cross\_refs\_SUPPORTED)

target\_link\_libraries(lib2 PRIVATE "$<LINK\_GROUP:cross\_refs,lib1,external>")

else()

target\_link\_libraries(lib2 PRIVATE lib1 external)

endif()

CMake will generate the following linker command line fragments when linking

``lib2``:

\* ``GNU``: ``-Wl,--start-group /path/to/lib1.a -lexternal -Wl,--end-group``

\* ``SunPro``: ``-Wl,-z,rescan-start /path/to/lib1.a -lexternal -Wl,-z,rescan-end``

Predefined Features

^^^^^^^^^^^^^^^^^^^

The following built-in group features are pre-defined by CMake:

.. include:: LINK\_GROUP\_PREDEFINED\_FEATURES.txt