BTC RevEngTools – Graph Introduction

Version: 0.2, last changed on 2010-Oct-26 by Simon Giesecke

Graph types

Technology-independent graphs

For any technology, a graph showing the link-time dependencies is generated, as well as a graph based on that which highlights violations of the EPM physical module dependency rules.

Overview and detail graphs

Each graph is output in an overview variant, where modules are grouped to module groups, and a detail variant, where each module is shown. An edge between two module groups in the overview variant means that there is at least one dependency between two modules contained in the module groups.

EPM Rule Violations

Note: The results may be inaccurate if the classification of the modules is not correct. By default, this is derived from naming conventions. The EPM physical module dependency rules are still under discussion.

C++-specific graphs

For C++, link-time dependencies are not guaranteed to be consistent with compile-time dependencies (i.e. include dependencies). There are additional graphs depicting the module-level include dependencies (with and without external dependencies). In addition, the consistency of compile-time and link-time dependencies is checked, and shown as a graph and a report. The results may be inaccurate if headers are not correctly associated with modules.

System-specific additions

CoreAssetBase

For the CoreAssetBase, the following conventions apply, if not noted otherwise:

- BtcCommonsCore is omitted (since nearly every other module depends on it),
- Test, configurator and marshaling modules are omitted, if they have only outgoing dependencies

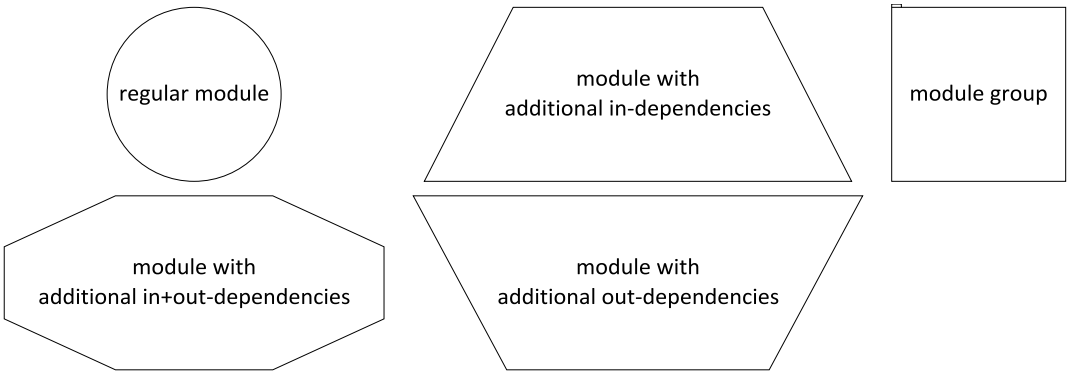
PRINS

- “PRINS legacy”, test, configurator and marshaling modules are omitted, if they have only outgoing dependencies
- The subgraph of modules (edges in the subgraphs, and the outline of nodes) that depend directly or indirectly on any PRINS legacy module is colored in light purple.

Common graph elements

Nodes

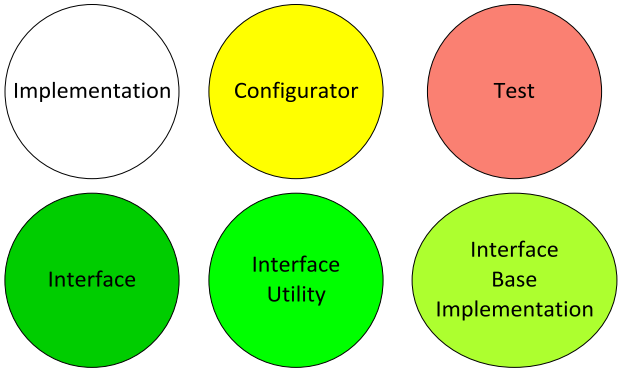
Shapes



“Additional” refers to the full dependency graph, i.e. these modules have additional dependencies that are not contained in the shown subgraph or connect to module types that are not shown.

Fill colors

The fill colors are based on the EPM module classification:



Edges

Edges forming a cycle are highlighted in red, and they are labelled with the number of the non-trivial strongly connected component (SCC) they belong to. The numbering of the SCCs is not stable across graphs.

Colors