3.17 What is new in MCAPI 2.000

The changes from version 1.0 to 2.000 are in the areas of improved API consistency, bug fixes, enhancements an added functionality.

3.17.1 MCAPI to MCA types

Some MCAPI data types and definitions were promoted to MCA data types and definitions to simplify interoperability between MCAPI and other MCA standards and to allow implementations to provide MCA versions of some functions that could be used for multiple MCA standards. For this purpose an mca .h header file was added.

3.17.2 MCAPI Domains

An MCAPI domain is comprised of one or more MCAPI nodes in a multicore topology and used for routing purposes. The scope of a domain is implementation defined and its scope could for example be a single chip with multiple cores or multiple processor chips on a board. The domain id is specified once at node initialization. A domain can contain multiple nodes. Some example potential uses for domains are topologies that may change dynamically, include non-MCAPI sub-topologies, require separation between different transports or have open and secure areas.

3.17.3 MCAPI endpoint attributes

A few attributes were added, for details see the attribute section for details. A certain number of endpoint attributes were reserved for MCA. Vendors desiring to add additional endpoint attributes in their implementation can request to be assigned a range of vendor specific endpoint attribute numbers from the MCA. Attribute ranges are defined in mca.h

3.17.4 MCAPI header files

The header files have been restructured to better separate standard and implementation specific definitions.

3.17.5 New functionality and functions

3.17.5.1 Initialization parameters and information

Initialization parameters were added to allow implementations to configure the MCAPI at run time. A parameter was also added to allow implementations to provide information about the MCAPI runtime with both MCAPI specified and implementation specific information. For specifics on MCAPI information see below. The topology information can for example be used for basic initial system discovery by using the `number_of_domains` and `number_of_nodes` parameters to establish communications for more thorough discovery.