

3.11 MRAPI Timeout/Cancellation Philosophy

The MRAPI API provides timeout functionality for its non-blocking calls through the timeout capability of the `mrapi_wait()` and `mrapi_wait_any()` functions. All blocking functions implementations have `timeout_t` parameters. Setting the timeout to 0 means a function call will not time out. Setting it to `MRAPI_INFINITE` means it will eventually timeout but only after the maximum number of tries.

The MRAPI API also provides cancellation functionality for its non-blocking calls through the `mrapi_cancel()` function.

3.12 MRAPI Datatypes

MRAPI uses predefined data types for maximum portability. The predefined MRAPI data types are defined in the following subsections. To simplify the use of multiple MCA (Multicore Association) standard API's some MRAPI data types have MCA equivalents and some MRAPI functions will have MCA equivalent functions that can be used for multiple MCA API's. An MRAPI implementation is not required to provide MCA equivalent functions.

In general, API parameters that refer to MRAPI entities are *opaque handles* that should not be examined or interpreted by the application program. Obtaining a handle is done either via a *create* function or a *get* function. Create/get functions require MRAPI *ID* types (see Sections 3.12.1, 3.12.2, 3.12.4, 3.12.6, 3.12.13) to be passed in and will return a handle (see Sections 3.12.5, 3.12.7, 3.12.8, 3.12.10, 3.12.11) for use in all other function calls related to that MRAPI object.

3.12.1 `mrapi_domain_t`

The `mrapi_domain_t` type is used for MRAPI domains. The domain id scheme is implementation defined. For application portability we recommend using symbolic constants in your code. The `mrapi_domain_t` has an `mca_domain_t` equivalent.

3.12.2 `mrapi_node_t`

The `mrapi_node_t` type is used for MRAPI nodes. The node numbering is implementation defined. For application portability we recommend using symbolic constants in your code. The `mrapi_node_t` has an `mca_node_t` equivalent.

3.12.3 Initialization parameters and information

Initialization parameters were added to allow implementations to configure the MRAPI runtime. A parameter was also added to allow implementations to provide information about the MRAPI runtime both MRAPI specified and implementation specific information. For specifics on RCAPAPI information see below.

3.12.3.1 `mrapi_param_t`

Initialization parameters will vary by implementation, and may include specifications of the amount of resources to be used for a specific implementation or configuration, such as for example the maximum number of nodes, etc.

3.12.3.2 `mrapi_info_t`

The informational parameters include MRAPI specified information as outlined below, as well as implementation specific information. Implementation specific information must be documented by the implementer.