rfoMM v0.20 API Cheat Sheet

Quick reference for functions available in rfoMM.bas.

Function Index

Initialization:
- MM_INIT()

User Functions:
- MM_CREATE(TYPE$)
- MM_CREATE$(TYPE$)
- MM_KEYS(MM_ADDRESS$)
- MM_KEYS$(MM_ADDRESS$)
- MM_TYPE$(MM_ADDRESS$)
- MM_DELETE(MM_ADDRESS$)
- MM_EXISTS(MM_ADDRESS$)
- MM_EXISTS$(MM_ADDRESS$)
- MM_POINTER(MM_ADDRESS$)
- MM_VALIDATE$(MM_ADDRESS$)
- MM_LOAD$(STRUCT_ID, TYPE$)
- MM_ADDRESS$(STRUCT_ID, TYPE$)

Internal Functions (Advanced/Low-Level):
- MM_OPEN(TYPE$)
- MM_RECYCLE(TYPE$)

Initialization

This function MUST be called once before any other rfoMM functions are used.

- **MM_INIT()**
  - **Args:** (None)
  - **Returns:** (None)
  - *Desc: Initializes the memory management environment, internal tracking bundles, and recycling lists.*

User Functions

Functions intended for general use by developers for creating and managing structures.

Core Management & Creation

- **MM_CREATE$(type$)**
  - **Args:** type$ (String: "B", "SL", "NL", "C", "O") - Type of structure to create.
  - **Returns:** MM_ADDRESS$ (String: e.g., "rfoMM(x)") - Unique address for the new structure.
  - *Desc: Creates a new managed structure, returning its address. Not meant as developer to create “C” or “O”, that is left to rfOOP commands.*

- **MM_CREATE(type$)**
  - **Args:** type$ (String: "B", "SL", "NL", "C", "O") - Type of structure to create.
  - **Returns:** Pointer (Numeric) - RFO Basic numeric pointer for the new structure.
  - *Desc: Wrapper for MM_CREATE$; creates structure and returns its raw pointer. Not meant as developer to create "C" or "O", that is left to rfOOP commands.*
- **MM_DELETE(MM_ADDRESS$)**
  - **Args:** MM_ADDRESS$ (String) - Address of the structure to delete/recycle.
  - **Returns:** (None)
  - *Desc: Removes structure from management, clears it, makes pointer available for recycling.*
- **MM_LOAD$(struct_id, type$)**
  - **Args:** struct_id (Numeric: >0), type$ (String: 'B','SL','NL', 'C', 'O') - Pointer and type of existing RFO structure.
  - **Returns:** MM_ADDRESS$ (String) or "" (Empty if struct_id < 1).
  - *Desc: Brings an existing RFO structure under MM management.*

Information & Retrieval

- **MM_POINTER(MM_ADDRESS$)**
  - **Args:** MM_ADDRESS$ (String) - Address of a managed structure.
  - **Returns:** Pointer (Numeric) - The underlying RFO Basic numeric pointer, or 0 if address not found.
  - *Desc: Gets the raw pointer associated with an address.*
- **MM_TYPE$(MM_ADDRESS$)**
  - **Args:** MM_ADDRESS$ (String) - Address of a managed structure.
  - **Returns:** TypeString$ (String: "B", "SL", "NL", "C", "O", or "UNDEFINED") - The type of the structure.
  - *Desc: Gets the type string associated with an address.*
- **MM_EXISTS(MM_ADDRESS$)**
  - **Args:** MM_ADDRESS$ (String) - Address to check.
  - **Returns:** Exists (Numeric: 1 for true, 0 for false)
  - *Desc: Checks if an address is currently managed.*
- **MM_EXISTS$(MM_ADDRESS$)**
  - **Args:** MM_ADDRESS$ (String) - Address to check.
  - **Returns:** Exists$ (String: "TRUE" or "FALSE")
  - *Desc: Checks if an address is currently managed (string version).*
- **MM_ADDRESS$(struct_id, type$)**
  - **Args:** struct_id (Numeric: >0), type$ (String: 'B','SL','NL', 'C', 'O') - Pointer and type of structure.
  - **Returns:** MM_ADDRESS$ (String) or "" (Empty string if not found or invalid input).
  - *Desc: Finds the existing address for an already-managed structure.*

Structure Specific Operations

- **MM_KEYS$(MM_ADDRESS$)**
  - **Args:** MM_ADDRESS$ (String) - Address of a managed Bundle.
  - **Returns:** KeysListAddress$ (String: MM_ADDRESS$ of a new String List).
  - *Desc: Gets Bundle keys, returns address of a new, managed String List containing*

the keys.

- **MM_KEYS(MM_ADDRESS$)**
  - **Args:** MM_ADDRESS$ (String) - Address of a managed Bundle.
  - **Returns:** KeysListPointer (Numeric) - Pointer to a new String List containing the keys.
  - *Desc: Wrapper for MM_KEYS$; returns the raw pointer of the new keys list.*
- **MM_VALIDATE$(MM_ADDRESS$)**
  - **Args:** MM_ADDRESS$ (String) - Address of a structure (likely Bundle intended as Class/Object).
  - **Returns:** Status$ (String: "TRUE" or "FALSE")
  - *Desc: Validates if the underlying bundle has expected rfOOP structure keys.*

Internal Functions (Advanced/Low-Level)

These functions are used internally by rfoMM. Direct use is generally not required or recommended.

- **MM_OPEN(type$)**
  - **Args:** type$ (String: "B", "SL", "NL") - Type of structure to check for recycling.
  - **Returns:** Pointer (Numeric) - A recycled numeric pointer of the specified type if available, otherwise 0.
  - *Desc: Checks if a previously deleted structure pointer is available for reuse.*
- **MM_RECYCLE(type$)**
  - **Args:** type$ (String: "B", "SL", "NL") - Type of structure needed.
  - **Returns:** Pointer (Numeric) - A recycled numeric pointer, removing it from the available pool, or creates a new one if none were recycled.
  - *Desc: Retrieves a recycled pointer for use, or triggers creation of a new one.*