Additional MiApp function set only applicable to MiWi Mesh 2.10 and above:

Function Type: Initialization

1) The function **MiApp_SetAddressPan** sets the device address and the PANID and has no return value. The input parameters to this function are short address of the node and the PANID. The short address will be a unique address for each wireless node in the network. PANID value should be the same for all the wireless nodes in a chosen network. This function should be used by all device roles – Pan Coordinator, Coordinator, and Sleeping RFD's.

```
void MiApp_SetAddressPan (uint8_t *Address, uint16_t Panid)
```

The function call MiApp_SetAddressPan returns no value. The function has two parameters.

- Address: This input parameter is the short address input for wireless node. For each node this address should be unique.
- Panid: This input parameter is the 2 byte network identifier (PANID) on which the wireless node will operate on.

Note: This function will be used only by Coordinator.

2) The function **MiApp_InitSleepRFDBuffers** is used to initialize the indirect message buffer for the devices in network. Indirect message buffers will define the number of messages cached inside the Parent device. The parent device for a RFD sleeping device could be Pan Coordinator or a Coordinator. Hence this function is only applicable for Pan Coordinator or Coordinator devices in network.

The function call MiApp_InitSleepRFDBuffers returns number of sleeping RFD buffers allocated. The function has three input parameters

- Buffer: This input parameter is pointer to the buffer array.
- BufferSize: This input parameter is the size of the buffer.
- rfdMaxDataSize: This input parameter is the maximum length of data to be stored for each Sleeping RFD's in the network.

```
uint16_t MiApp_InitSleepRFDBuffers (uint8_t *Buffer, uint16_t
BufferSize, uint16_t rfdMaxDataSize);
```

Note: The macro ENABLE_INDIRECT_MESSAGE should be enabled in the Pan Coordinator for the function to execute.

Function Type: Status

3) The function MiApp_GetParentAddress is used by Coordinator or a Sleeping RFD to identify their parent's short address. Usually wireless mesh network consists of multiple nodes connected to each other,in order to understand the network mapping, it is important to know the relationship between the nodes.

```
addr_t MiApp_GetParentAddress (void)
```

Input:

The function **MiApp_GetParentAddress** has no input parameter and returns an address. All addresses that are received to the application layer are stored in a global variable defined in the format of addr_t as follows:

```
typedef struct addr_t_def
{
   uint8_t bytes[ADDRESS_LEN];
}addr_t;
```

addr_t: Structure to store the address of the parent node to which the Coordinator or RFD Sleeping device is connected as a child.

Note: This function will be used only by Coordinator or a Sleeping RFD.

4) In a Mesh Network if a wireless node loses its connection with its parent node or if it is not part of network, there should be a way to find out the lost connection from the application layer. The function **MiApp_IsMemberOfNetwork** can be used to identify whether a network exists and the node is part of the existing network. There is no input parameter for this function and will return a boolean true if the node is part of an existing network. This function when called by the Pan Coordinator device will always return a "true" value (as Pan Coordinator creates the network). This function when used in a RFD Sleeping Device or a Coordinator will help find the status of wireless node in the network. If a boolean "false" is returned by the function call it indicates that the node is not part of any network.

```
bool MiApp_IsMemberOfNetwork (void)
```

Function Type: Receiving Messages

5) In a wireless network when RFD sleeping devices are used, there needs to be a mechanism for these devices to wake up and receive a data packet. The function **MiApp_RequestData** is used by RFD sleeping devices to request for cached data from their parent device.

```
void MiApp_RequestData (void)
```

This function has no input parameter and returns no value. *rxMessage.SourceAddress* holds the received data.

Note: This function is only applicable for Sleeping RFD devices.

Type: Special Function

6) The function MiApp_SetNetworkSecure is used to secure all the messages in network.
The following packet types will be secured:
   - Data packet.
   - Command packets.
   - Status packets.
   - Control packets.
   - Network maintenance.

This function should be called by all nodes (Pan Coordinator, Coordinator, RFD Sleeping Device) in the network. This function has no input parameter and returns no value.

```
void MiApp_SetNetworkSecure (void)
```