

# Using interfaces for Brew MP 1.0.2 and lower

[brewmp.com](https://developer.brewmp.com/resources/tech-guides/position-determination-technology-guide/using-interfaces/using-interfaces-brew-mp-102-and-lower) |

Using interfaces for Brew MP 1.0.2 and lower

This section discusses how to use the position determination interfaces available in Brew MP 1.0.2 and below.

Invoking a position fix using IPosDet

Follow these steps to request a position fix using IPosDet:

1. Configure the GPS engine using IPosDet\_SetGPSConfig (optional).
2. Request a position fix using IPosDet\_GetGPSInfo.
3. Process the results (IPosDet Callback).

Note: Requesting a position fix using IPosDet\_GetGPSInfo without configuring the GPS engine uses the default configuration.

To configure the position determination engine based on the needs of the application, call IPosDet\_SetGPSConfig() and set the parameters of the AEEGPSConfig data structure.

For example, a turn-by-turn navigation application would most likely set the Mode parameter to AEEGPS\_MODE\_TRACK\_LOCAL to force the positioning engine into MS-based mode for the fix requests.

See the C/C++ API Reference on the Brew MP website for more information on the IPosDet\_SetGPSConfig() return codes.

The parameters in the AEEGPSConfig data structure are:

* **Mode** specifies which operating mode the underlying GPS engine uses.
* **Optim** specifies whether to optimize the position determination request for speed, accuracy, or data exchange
* **QoS** corresponds to the number of seconds that the application wants the position determination engine to search for satellites
* **Server** specifies the server type and configuration
* **nFixes** estimated number of repetitive position determination requests the interface will make when in Track mode.
* **nInterval** estimated interval between consecutive position determination requests, in seconds.

See AEEGPSConfig in the C/C++ API Reference on the Brew MP website for more information.

After configuring the location engine, call IPosDet\_GetGPSInfo() to obtain a position fix:

Note: IPosDet\_GetGPSInfo() is an asynchronous function that uses the AEECallback and AEEGPSInfo structures. Ensure that the callback and information structures passed to this method remain in scope until the callback returns.

Completing a requested position determination triggers the callback function. The position response fills the AEEGPSInfo structure.

Note: To implement a position tracking application that periodically requests position information, wait for the callback to return before you initiate a new IPosDet\_GetGPSInfo() call. IPosDet\_GetGPSInfo() expects certain input parameters. See the C/C++ API Reference on the Brew MP website for more information on the IPosDet\_GetGPSConfig() return codes.

The following code snippet explains how to convert the WGS84 latitude/longitude values into degrees, where pMe->m\_GPSInfo.dwLat and pMe->m\_GPSInfo.dwLon are filled with WGS84 values for latitude and longitude values, respectively, and 186413.5111 is the WGSFactor:

To determine the validity of information received for a specific field, applications can filter it using the appropriate mask. For example, to verify the validity of the latitude information:

IPosdet callback triggers when the GPS request completes. The callback function checks for the status of the request. If the status is AEEGPS\_ERR\_NO\_ERR, the request is successful; otherwise, the request returns an error code. See the C/C++ API Reference on the Brew MP website for more information.

Accuracy parameter for IPosDet\_GetGPSInfo()

The IPosDet\_GetGPSInfo() accuracy parameter is only used in fall-back scenarios (e.g., Track\_Optimal mode with speed optimization). The accuracy parameter indicates an accuracy level from 1 to 6; level 1 is the the default and lowest accuracy level (AEEGPS\_ACCURACY\_LEVEL1), and level 6 is the highest accuracy level (AEEGPS\_ACCURACY\_LEVEL6).

These accuracy levels correspond to accuracy threshold values in meters of 1000, 500, 250, 100, 50, 25 for GPS fixes. For instance, a GPS fix requested with AEEGPS\_ACCURACY\_LEVEL1 obtains the fix values with a position uncertainty close to 1000m; a fix requested with AEEGPS\_ACCURACY\_LEVEL6 obtains a fix with vertical uncertainty close to 25m.

Usually, a more accurate fix takes more time to compute than a less accurate fix, so, a GPS fix is faster for the AEEGPS\_ACCURACY\_LEVEL1 than the AEEGPS\_ACCURACY\_LEVEL6. A high value of accuracy threshold, such as 1000m or AEEGPS\_ACCURACY\_LEVEL1, may produce poor performance in many GPS engines; the lower the accuracy threshold, the better the performance.

See AEEGPSAccuracy in the C/C++ API reference on the Brew MP website for more information on the accuracy parameter.

Sector-based position location

Sector-based position location is supported on CDMA devices only. Calling the following function returns sector-based location information such as the SystemID, NetworkID, BaseStationID, BaseStationClass, and best Pilot.

|  |  |
| --- | --- |
| 1 | int IPOSDET\_GetSectorInfo(IPOSDET \* pif, |

2AEESectorInfo \* pSecInfo) This information is proprietary to operators. The use of the sector-based position determination function, IPOSDET\_GetSectorInfo(), requires the PL\_SECTORINFO sector information privilege.

Placing a single request for GPS position information

This sample is useful for applications requesting position information to:

* display a map with the current position indicated by a mark
* display a list of services available near-by
* similar tasks.

The information returned is GPS-based position information. The sequence of Brew MP function calls is illustrated in the following diagram:

Establish the type of position fix by calling IPOSDET\_SetGPSConfig() and passing the AEEGPSConfig structure as a parameter. To simplify the configuration process, applications should only modify the configuration fields of interest. The existing configuration can be obtained using IPOSDET\_GetGPSConfig().

A position fix can be obtained without configuring the position determination engine if the default configuration is acceptable. The following example sets only one possible configuration item (see AEEGPSConfig and IPOSDET\_SetGPSConfig() in the C/C++ API Reference on the Brew MP website to determine what configuration is appropriate for a particular application).

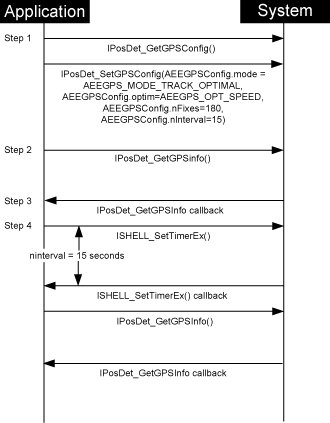
Ensure that the callback and information structures passed to this function remain in scope until the callback returns. For example:

The position response is placed in the memory pointing to the structure AEEGPSInfo. The status field of the AEEGPSInfo structure indicates whether the position determination was successful. For example:

AEEGPSInfo provides position information in the IS-801 location response format. In this format, latitude, longitude, velocity, and altitude are provided as fixed point values. Applications can convert these values to floating point degrees before using them. IPOSDET\_ExtractPositionInfo() is a conversion utility that converts latitude, longitude, velocity, and altitude from fixed point to floating point degrees. Uncertainty is converted from enumerated Gaussian distribution, as specified by IS-801, to meters. For example: Placing multiple requests for GPS position information

This sample is useful for applications requesting position information to

* track a current position on a map
* give turn-by-turn instructions with recalculation when the user goes off course, or
* similar tasks.

The information returned in this sample is GPS-based position information. To implement a position tracking application that requests positions periodically, wait for the callback to return and then initiate a new IPOSDET\_GetGPSInfo() call. The sequence of Brew MP function calls is illustrated in the following diagram:

The sample for placing a single request for GPS position information contains more details on the basics of configuring the position determination engine. Applications that are making multiple position determination requests may specify the AEEGPSConfig mode parameter as AEEGPS\_MODE\_TRACK\_OPTIMAL. To perform tracking, the application must call the IPOSDET\_GetGPSInfo() function at regular intervals.The AEEGPSConfig optim parameter is used to tell the position determination engine the desired optimization. Applications may:

* optimize for speed
* optimize for accuracy
* minimize data transfer with the server.

Ensure that the callback and information structures passed to this function remain in scope until the callback returns. For example:

IPOSDET\_GetGPSInfo() enforces the privacy policies recommended by the network operator. Handling privacy dialog events is described in step 3 of the sample titled " Placing a single request for GPS position information". See step 4 (steps 3 and step 4 are shown together in the following sample).

The position response is placed in the memory pointing to the structure AEEGPSInfo. The status field of the AEEGPSInfo structure indicates if position determination was successful. Applications may convert the data in the AEEGPSInfo structure to the AEEPositionInfoEx format. The use of IPOSDET\_ExtractPositionInfo() to accomplish this is described in step 5 of the Placing a single request for GPS position information sample. A tracking application can initiate a new position request once the IPOSDET\_GetGPSInfo() callback has been called. A timer can be used to initiate the new request after the appropriate interval.

The new position determination request is placed from the timer callback function. For example:

You can download and modify the IPosDet sample to become familiar with the IPosDet capabilities (see the sample code associated with this guide on the Brew MP website).

Sample code for placing single and multiple requests

©2011-2015 Qualcomm Technologies, Inc. All rights reserved. Legal | Privacy Policy | Sitemap