Interface documentation of HCM-MS (C#)

Table of Contents

General information

The `HCMMS_CS.dll` is a dynamic link library written in `C#` thus using Common Intermediate Language (CIL). This allows our assembly to be referenced by all CLI languages such as `C++/CLI`, `F#`, `Swift`, `IronPython` or `C#` itself (as long as the used data types are compatible). To use it within other than CLI languages, a wrapper is necessary. This can be achieved with our second assembly `HCMMS_UMB.dll`. A brief description is given later on.

Loading the DLL

For loading and calling the DLL there are two possible ways: Implicit linking (also called static load or load-time dynamic linking) and explicit linking (also called dynamic load or run-time dynamic linking). The former provides the advantage of type checking at compile time while the latter allows the DLL to be easyily exchangable without rebuilding the project.

Implicit linking

Using implicit linking the DLL will be loaded at the same time as the executable that uses it. The exported function will be called as if the function were statically linked and contained within the executable. To use implicit linking the DLL has to be referenced by the project. For example to do so within Visual Studio 2017 right-click on the `References` item in your project in `Solution Explorer`, then use the `Browse` button in the `Add Reference` dialog (in newer versions `Project`, `Add reference`, `Browse`). The function `HCMMS_V7_DLL()` can then be called using the namespace `HCM.DataObj` and class `ILegacy` like that:

```
HCM.DataObj.ILegacy.HCMMS_V7_DLL(ref C_mode, ref Bor_dis, ref PD, ref Distance, ref H_Datab_Tx, ref
H_Datab_Rx, ref HCM_error, ref Heff, ref Dh, ref Dh_corr, ref Power_to_Rx, ref Free_space_FS, ref Land_FS, ref
Sea_FS, ref tx_Ant_corr, ref tx_Ant_type_corr, ref Dir_tx_Rx, ref V_angle_tx_Rx, ref tx_TCA, ref rx_TCA, ref
tx_TCA_corr, ref rx_TCA_corr, ref D_sea_calculated, ref rx_Ant_corr, ref rx_Ant_type_corr, ref
Delta_frequency, ref Corr_delta_f, ref Calculated_FS, ref Perm_FS, ref CBR_D, ref ERP_ref_Tx, ref Prot_margin,
ref I_str);
```

Explicit linking

Explicit linking as known by the former Fortran-90 DLL might be implemented in a later version.

Calling the DLL function

To see a detailed description on how to call the function and of all parameters as well as error codes see the corresponding documentation page.

Class ILegacy

Inheritance

System.Object

ILegacy

Inherited Members

System.Object.ToString()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: HCM.DataObj

Assembly: HCMMS_CS.dll

Syntax

```
public static class ILegacy
```

Methods

HCMMS_V7_DLL(ref Int32, ref Int32, ref Double, ref Double, ref Int16, ref Int16, ref Int32, ref Single, ref Single, ref Single, ref Single, ref Single, ref Single, ref Single, ref Single, ref Single, ref Double, ref Double, ref Single, ref Single, ref Single, ref Single, ref Double, ref Single, ref Single, ref Double, ref Single, ref Single, ref Single, ref Single, ref Single, ref Single, ref String)

The main interface to the HCM Dll according to the official Fortran interface.

Declaration

```
public static void HCMMS_V7_DLL(ref int C_mode, ref int Bor_dis, ref double PD, ref double Distance, ref short H_Datab_Tx, ref short H_Datab_Rx, ref int HCM_error, ref float Heff, ref float Dh, ref float Dh_corr, ref float Power_to_Rx, ref float Free_space_FS, ref float Land_FS, ref float Sea_FS, ref float tx_Ant_corr, ref float tx_Ant_type_corr, ref double Dir_tx_Rx, ref double V_angle_tx_Rx, ref float tx_TCA, ref float rx_TCA, ref float tx_TCA_corr, ref float rx_TCA_corr, ref double D_sea_calculated, ref float rx_Ant_corr, ref float rx_Ant_type_corr, ref double Delta_frequency, ref float Corr_delta_f, ref float Calculated_FS, ref float Perm_FS, ref float CBR_D, ref float ERP_ref_Tx, ref float Prot_margin, ref string I_Str)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	C_mode	[INPUT] The mode of calculation
System.Int32	Bor_dis	[INPUT] The distance to the borderline
System.Double	PD	[INPUT] The distance between two profile points
System.Double	Distance	[OUTPUT] The distance between transmitter and the receiving point in km

TYPE	NAME	DESCRIPTION
System.Int16	H_Datab_Tx	[OUTPUT] The height of the transmitter site above sea level from the terrain database in m
System.Int16	H_Datab_Rx	[OUTPUT] The height of the receiver site above sea level from the terrain database in m
System.Int32	HCM_error	[OUTPUT] The error code. Error codes are listed in the HCM documentation. Additional error codes are listed in remarks.
System.Single	Heff	[OUTPUT] The effective antenna height in m used for the calculations according the ITU_R method.
System.Single	Dh	[OUTPUT] The terrain irregularity in m used for the calculations according the ITU_R method.
System.Single	Dh_corr	[OUTPUT] The correction factor according to the terrain irregularity in dB used for the calculations according the ITU_R method.
System.Single	Power_to_Rx	[OUTPUT] The power in the direction of the receiver in dBW.
System.Single	Free_space_FS	[OUTPUT] The free space field strength in dBµV/m.
System.Single	Land_FS	[OUTPUT] The land field strength in dBµV/m.
System.Single	Sea_FS	[OUTPUT] The sea field strength in dBµV/m.
System.Single	tx_Ant_corr	[OUTPUT] The correction factor in dB according to the transmitter antenna type (horizontal and vertcal).
System.Single	tx_Ant_type_corr	[OUTPUT] The correction factor in dB according to the transmitter antenna type ('E' or 'I').
System.Double	Dir_tx_Rx	[OUTPUT] The horizontal direction from the transmitter to the receiver in degrees.
System.Double	V_angle_tx_Rx	[OUTPUT] The vertical direction from the transmitter to the receiver in degrees.
System.Single	tx_TCA	[OUTPUT] The transmitter clearance angle in degrees.

TYPE	NAME	DESCRIPTION
System.Single	rx_TCA	[OUTPUT] The receiver clearance angle in degrees.
System.Single	tx_TCA_corr	[OUTPUT] The correction factor according to the transmitter clearance angle in dB.
System.Single	rx_TCA_corr	[OUTPUT] correction factor according to the receiver clearance angle in dB.
System.Double	D_sea_calculated	[OUTPUT] The distance over sea in km taken into account during the calculations (either input value or calculated value).
System.Single	rx_Ant_corr	[OUTPUT] The correction factor in dB according to the receiver antenna type (horizontal and vertical).
System.Single	rx_Ant_type_corr	[OUTPUT] The correction factor in dB according to the receiver antenna type ('E' or 'I').
System.Double	Delta_frequency	[OUTPUT] The frequency difference between transmitter - and receiver frequency in kHz.
System.Single	Corr_delta_f	[OUTPUT] The correction factor according to the frequency difference between transmitter - and receiver frequency in dB.
System.Single	Calculated_FS	[OUTPUT] The calculated field strength in dBµV/m.
System.Single	Perm_FS	[OUTPUT] The permissible field strength in dBµV/m (input value or calculated value).
System.Single	CBR_D	[OUTPUT] The maximum cross border range in km (input value or from Agreement).
System.Single	ERP_ref_Tx	[OUTPUT] The power of the reference transmitter in dBW.
System.Single	Prot_margin	[OUTPUT] The protection margin in dB (difference of calculated field strength and permissible field strength).
System.String	I_Str	[INPUT/OUTPUT] A String variable with at least 432 characters. Content is described in remarks.

Remarks

Error Codes:

- 0 no error
- 36 error opening terrain- or morphological data file (data not available)
- 200 error in longitude (in 'Point_heighe or 'Point_type' subroutine)
- 210 error in latitude (in 'Point height' or 'Point_type' subroutine)
- 220 error reading record (in 'Point_heighf or 'Point_type' subroutine)
- 300 latitude is not in range of 0.0 - 90.0 (in 'Point_height' or 'Point_type' ! subroutine)
- 400 height is missing (-9999) (in 'Point height' subroutine)
- 1000 Distance between Tx and Rx = 0. Calculations not possible
- 1001 Error in geographical coordinates (Tx longitude, degrees)
- 1002 Ener in geographical coordinates (Tx longitude, minutes)
- 1003 Error in geographical coordinates (Tx longitude, seconds)
- 1004 Error in geographical coordinates (Tx longitude, E/W)
- 1005 Error in geographical coordinates (Tx latitude, degrees)
- 1006 Error in geographical coordinates (Tx latitude, minutes)
- 1007 Error in geographical coordinates (Tx latitude, seconds)
- 1008 Error in geographical coordinates (Tx latitude, N/S)
- 1009 Error in Tx antenna height
- 1010 Error in transmitting frequency value
- 1011 Error in transmitting frequency unit
- 1012 Error in radius of service area of Tx
- 1013 Error in input value height of Tx site above sea level
- 1014 Error in geographical coordinates (Rx longitude, degrees)
- 1015 Error in geographical coordinates (Rx longitude, minutes)
- 1016 Error in geographical coordinates (Rx longitude, seconds)
- 1017 Error in geographical coordinates (Rx longitude, E/W)
- 1018 Error in geographical coordinates (Rx latitude, degrees)
- 1019 Error in geographical coordinates (Rx latitude, minutes)
- 1020 Error in geographical coordinates (Rx latitude, seconds)
- 1021 Error in geographical coordinates (Rx latitude, N/S)
- 1022 Error in Rx antenna height
- 1023 Error in reception frequency value
- 1024 Error in reception frequency unit
- 1025 C_mode is out of range
- 1026 Error in input value of permissible field strength
- 1027 Error in input value of maximum cross border range
- 1028 The distance is greater than 1000 km. Calculations not possible.
- 1029 Error in radius of Rx service area
- 1030 Error in input value Rx site height above sea level
- 1031 Error in Tx elevation
- 1032 Error in Tx azimuth
- 1033 Error in type of Tx antenna ("E" or "I")
- 1034 Error in power
- 1035 Error in input value of distance over sea
- 1036 The 'xxx.ALL' borderline file for Tx is missing
- 1037 The 'xxx.ALL' borderline file for Rx is missing
- 1038 Error in type of antenna (TR20-08)
- 1039 Error in input data of correction factor according frequency difference
- 1040 Channel spacing outside definition range (Rx)
- 1041 Channel spacing outside definition range (Tx)
- 1042 Error in Rx elevation
- 1043 Error in Rx azimuth
- 1044 Error in Rx type of antenna ("E" or "I")

- 1045 Error in gain of Rx antenna
- 1046 Error in input data of depolarization loss
- 1047 Distance to borderline is too long
- 1048 Selected line data not available
- 1049 Error in line data
- 1050 No Agreement frequency and CBR input or PermFS is missing
- 2000 wrong Figure frequency (from Get_figure_FS_value)
- 2001 wrong Time_percentage (from Get_figure_FS_value)
- 2002 wrong Sea_temperature (from Get_figure_FS_value)
- 2003 wrong Figure_Heff (from Get_figure_FS_value)
- 2004 wrong Figure_distance (from Get_figure_FS_value)
- 3000 DLL inputstring too short
- -1 unexpected Error
- -2 HCM DLL not found
- -3 HCM DLL entry point not found

I_Str definition:

- Pos. 1-8: Tx co-ordinates (E/W)
- Pos. 9-15: Tx co-ordinates (N/S)
- Pos. 16-23: Rx co-ordinates (E/W)
- Pos. 24-30: Rx co-ordinates (N/S)
- Pos. 31-34: Height of Tx site
- Pos. 35-38: Height of Rx site
- Pos. 39-45: Tx hor. ant. Type
- Pos. 46-52: Tx vert. ant. Type
- Pos. 53-57: Tx Azimut
- Pos. 58-62: Tx Elevation
- Pos. 63-66: Tx Antenna height
- Pos. 67-70: Rx Antenna height
- Pos. 71-71: Tx Type of Antenna
- Pos. 72-77: Max. radiated Power
- Pos. 78-89: Tx Frequency
- Pos. 90-90: Channel occupation
- Pos. 91-91: Sea temperature
- Pos. 92-96: Tx Service Area
- Pos. 97-101: Rx Service Area
- Pos. 102-106: Distance over Sea
- Pos. 107-118: Rx Frequency
- Pos. 119-127: Rx Design of emm.
- Pos. 128-136: Tx Design of emm.
- Pos. 137-143: Rx hor. ant. Type
- Pos. 144-150: Rx vert. ant. Type
- Pos. 151-155: Rx Azimut
- Pos. 156-160: Rx Elevation
- Pos. 161-161: Rx Type of Antenna
- Pos. 162-165: Rx Antenna gain
- Pos. 166-169: Depolarization loss
- Pos. 170-174: Perm. Field strength
- Pos. 175-178: Corr.fact.acc.freq.diff.
- Pos. 179-181: Country code to calcul. to
- Pos. 182-184: County code of Tx

- Pos. 185-187: Max. cross boder range
- Pos. 188-250: Path of terrain data
- Pos. 251-313: Path of borderline data
- Pos. 314-376: Path of morphological data
- Pos. 377-382: OUTPUT: Version_number
- Pos. 383-402: OUTPUT: Info_values
- Pos. 403-417: OUTPUT: Tx_co_ordinates_calculated
- Pos. 418-432: OUTPUT: Rx_co_ordinates_calculated
- Pos. 433-???: Debug directory (optional)

Examples

An example of calling the dll function within C# using example values and constructing the input string manually.

```
// I/O parameters
int C_mode = 0;
int Bor_dis = 0;
double PD = 0.1;
double Distance = 0;
short H_Datab_Tx = 0;
short H_Datab_Rx = 0;
int HCM_error = 0;
float Heff = 0;
float Dh = 0;
float Dh_corr = 0;
float Power_to_Rx = 0;
float Free_space_FS = 0;
float Land_FS = 0;
float Sea_FS = 0;
float tx_Ant_corr = 0;
float tx_Ant_type_corr = 0;
double Dir_tx_Rx = 0;
double V_angle_tx_Rx = 0;
float tx_TCA = 0;
float rx_TCA = 0;
float tx_TCA_corr = 0;
float rx_TCA_corr = 0;
double D_sea_calculated = 0;
float rx_Ant_corr = 0;
float rx_Ant_type_corr = 0;
double Delta_frequency = 0;
float Corr_delta_f = 0;
float Calculated_FS = 0;
float Perm_FS = 0;
float CBR_D = 0;
float ERP_ref_Tx = 0;
float Prot_margin = 0;

// input string parameters
string tx_Longitude = "012E3743";
string tx_Latitude = "48N5203";
string rx_Longitude = "008E2352";
string rx_Latitude = "49N5112";
string tx_Height = "    ";
string rx_Height = "   0";
string tx_HorizontalAntennaType = "012EA05";
string tx_VerticalAntennaType = "012EA05";
string tx_Azimut = "200.0";
string tx_Elevation = " -2.0";
string tx_AntennaHeight = "  29";
string rx_AntennaHeight = "  10";
string tx_TypeOfAntenna = "E";
string tx_MaxRadiatedPower = " 14.22";
```

```
string tx_FrequencyValue = " 3655.00000";
string tx_FrequencyUnit = "M";
string ChannelOccupation = "1";
string SeaTemperature = "C";
string tx_RadiusServiceArea = "    0";
string rx_RadiusServiceArea = "    0";
string DistanceOverSea = "     ";
string rx_FrequencyValue = " 3655.00000";
string rx_FrequencyUnit = "M";
string rx_DesignationEmmission = "80M0G7WET";
string tx_DesignationEmmission = "80M0G7WET";
string rx_HorizontalAntennaType = "000ND00";
string rx_VerticalAntennaType = "000ND00";
string rx_Azimut = "  0.0";
string rx_Elevation = "  0.0";
string rx_TypeOfAntenna = "E";
string rx_AntennaGain = " 0.0";
string DepolarisationLoss = " 0.0";
string PermissibleFieldStrength = "  0.0";
string CorrectionFacorFrequencyDifference = "    ";
string LandTo = "D__";
string LandFrom = "D__";
string CrossBorderRangeMax = "   ";
string TopoPath = "D:\\TOPO                                                    ";
string BorderPath = "D:\\BORDER                                                  ";
string MorphoPath = "D:\\MORPHO                                                  ";
string OutVersionNumber = "      ";
string OutInfoValues = "                    ";
string OutTxCoordinatesCalculated = "               ";
string OutRxCoordinatesCalculated = "               ";

// composing of input string
string I_str = tx_Longitude + tx_Latitude + rx_Longitude + rx_Latitude + tx_Height + rx_Height +
tx_HorizontalAntennaType + tx_VerticalAntennaType + tx_Azimut + tx_Elevation + tx_AntennaHeight +
rx_AntennaHeight + tx_TypeOfAntenna + tx_MaxRadiatedPower + tx_FrequencyValue + tx_FrequencyUnit +
ChannelOccupation + SeaTemperature + tx_RadiusServiceArea + rx_RadiusServiceArea + DistanceOverSea +
rx_FrequencyValue + rx_FrequencyUnit + rx_DesignationEmmission + tx_DesignationEmmission +
rx_HorizontalAntennaType + rx_VerticalAntennaType + rx_Azimut + rx_Elevation + rx_TypeOfAntenna +
rx_AntennaGain + DepolarisationLoss + PermissibleFieldStrength + CorrectionFacorFrequencyDifference + LandTo +
LandFrom + CrossBorderRangeMax + TopoPath + BorderPath + MorphoPath + OutVersionNumber + OutInfoValues +
OutTxCoordinatesCalculated + OutRxCoordinatesCalculated;

// full composed input string
// string I_str = "012E374348N5203008E235249N5112       0012EA05012EA05200.0 -2.0  29  10E 14.22 3655.00000M1C
0    0      3655.00000M80M0G7WET80M0G7WET000ND00000ND00  0.0  0.0E 0.0 0.0  0.0    D__D__   D:\\TOPO
D:\\BORDER                                                  D:\\MORPHO
";

// HCM-DLL call
HCM.DataObj.ILegacy.HCMMS_V7_DLL(ref C_mode, ref Bor_dis, ref PD, ref Distance, ref H_Datab_Tx, ref
H_Datab_Rx, ref HCM_error, ref Heff, ref Dh, ref Dh_corr, ref Power_to_Rx, ref Free_space_FS, ref Land_FS, ref
Sea_FS, ref tx_Ant_corr, ref tx_Ant_type_corr, ref Dir_tx_Rx, ref V_angle_tx_Rx, ref tx_TCA, ref rx_TCA, ref
tx_TCA_corr, ref rx_TCA_corr, ref D_sea_calculated, ref rx_Ant_corr, ref rx_Ant_type_corr, ref
Delta_frequency, ref Corr_delta_f, ref Calculated_FS, ref Perm_FS, ref CBR_D, ref ERP_ref_Tx, ref Prot_margin,
ref I_str);
```

Interface for non-managed languages

To use the DLL outside of the managed world with languages such as `C` or `C++` it is necessary to use a wrapper. We provide `HCMMS_UMB.dll` which depends on our original assembly `HCMMS_CS.dll` in the same place and provides the exact same interface for non managed languages. For compatibility purposes a header file of `HCMMS_UMB.dll` is given below.

```
#pragma once
#include <string>

extern "C"
__declspec(dllexport)
void HCMMS_V7_DLL(int &C_mode, int &Bor_dis, double &PD, double &Distance, short int &H_Datab_Tx, short int
&H_Datab_Rx, int &HCM_error, float &Heff, float &Dh, float &Dh_corr, float &Power_to_Rx, float &Free_space_FS,
float &Land_FS, float &Sea_FS, float &Tx_ant_corr, float &Tx_ant_type_corr, double &Dir_Tx_Rx, double
&V_angle_Tx_Rx, float &Tx_TCA, float &Rx_TCA, float &Tx_TCA_corr, float &Rx_TCA_corr, double
&D_sea_calculated, float &Rx_ant_corr, float &Rx_ant_type_corr, double &Delta_frequency, float &Corr_delta_f,
float &Calculated_FS, float &Perm_FS, float &CBR_D, float &ERP_ref_Tx, float &Prot_margin, char I_str[], int
StrLen);
```

A short example based on the former example to call the function of the DLL using `C` and including the above mentioned header file follows on the next page.

```
#include <iostream>
#include "HCM_UMB.h"

int main()
{
    int C_mode = 0;
    int Bor_dis = 0;
    double PD = 0;
    double Distance = 0;
    short int H_Datab_Tx = 0;
    short int H_Datab_Rx = 0;
    int HCM_error = 0;
    float Heff = 0;
    float Dh = 0;
    float Dh_corr = 0;
    float Power_to_Rx = 0;
    float Free_space_FS = 0;
    float Land_FS = 0;
    float Sea_FS = 0;
    float Tx_ant_corr = 0;
    float Tx_ant_type_corr = 0;
    double Dir_Tx_Rx = 0;
    double V_angle_Tx_Rx = 0;
    float Tx_TCA = 0;
    float Rx_TCA = 0;
    float Tx_TCA_corr = 0;
    float Rx_TCA_corr = 0;
    double D_sea_calculated = 0;
    float Rx_ant_corr = 0;
    float Rx_ant_type_corr = 0;
    double Delta_frequency = 0;
    float Corr_delta_f = 0;
    float Calculated_FS = 0;
    float Perm_FS = 0;
    float CBR_D = 0;
    float ERP_ref_Tx = 0;
    float Prot_margin = 0;

    // full composed input string
    char I_str[] = "012E374348N5203008E235249N5112       0012EA05012EA05200.0 -2.0  29  10E 14.22
3655.00000M1C    0    0      3655.00000M80M0G7WET80M0G7WET000ND00000ND00  0.0  0.0E 0.0 0.0  0.0    D__D__
D:\\TOPO                                                    D:\\BORDER
D:\\MORPHO
"
    int StrLen = sizeof(I_str);

    // HCM-DLL call
    HCMMS_V7_DLL(C_mode, Bor_dis, PD, Distance, H_Datab_Tx, H_Datab_Rx, HCM_error,
        Heff, Dh, Dh_corr, Power_to_Rx, Free_space_FS, Land_FS, Sea_FS,
        Tx_ant_corr, Tx_ant_type_corr, Dir_Tx_Rx, V_angle_Tx_Rx, Tx_TCA, Rx_TCA,
        Tx_TCA_corr, Rx_TCA_corr, D_sea_calculated, Rx_ant_corr, Rx_ant_type_corr,
        Delta_frequency, Corr_delta_f, Calculated_FS, Perm_FS, CBR_D, ERP_ref_Tx,
        Prot_margin, I_str, StrLen);
}
```