

**NHDPlusV2 VAA Navigator
User Guide
Horizon Systems Corporation
12/11/2017**

Contents

NHDPlusV2 VAA Navigator Description	1
NHDPlusV2 VAA Navigator Databases	1
Using the NHDPlusV2 VAA Navigator Toolbar	2
Using the NHDPlusV2 VAA Navigator from a User-written Program	8

NHDPlusV2 VAA Navigator Description

The NHDPlusV2 VAA Navigator performs navigation on the NHDPlusV2 surface water network using NHDPlusV2 Value Added Attributes (PlusFlowlineVAA.dbf). The Navigator may be used through ArcMap via a special ArcMap toolbar or it may be called from user-written program code. The Navigator will work on any NHDPlusV2 workspace and requires the NHDSnapshot and NHDPlusAttributes components to be installed for each VPU on which the Navigator will be used.

The Navigator performs four types of navigation: upstream mainstem, upstream with tributaries, downstream mainstem, and downstream with divergences. Navigations can begin and end on full NHDFlowline features or may begin and end as points along features.

Any of the four types of navigation may be stopped based on a user-supplied distance from the starting point. Navigation results may also be filtered based on certain user specified criteria.

NHDPlusV2 VAA Navigator Databases

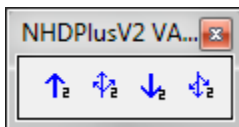
From the ArcMap toolbar, the first time the VAA Navigator is executed for a given NHDPlusV2 VPU, it builds a Microsoft SQL database for that VPU which speeds up future navigations. The database is called V02NavDB_<vvvvvvvv>.mdf, where <vvvvvvvv> is the VPUid for the NHDFlowline feature class being navigated. The database is created in a local folder specified by the user. The second and subsequent times that the Navigator is executed, it looks for the database and, if found, uses it.

Recommendation: In the navigation options dialog displayed when the VAA Navigator Toolbar is used, specify that the Navigator databases are to be stored with your installed NHDPlusV2 data in a folder that you create for that purpose. You must have read/write access to this folder.

Note: When called from a user-written program, the NHDPlusV2 VAA Navigator is capable of navigating drainage areas that contain multiple VPUs. When used through the ArcMap toolbar, only a single VPU will be navigated during each use of the tool.

Using the NHDPlusV2 VAA Navigator Toolbar

1. Description:



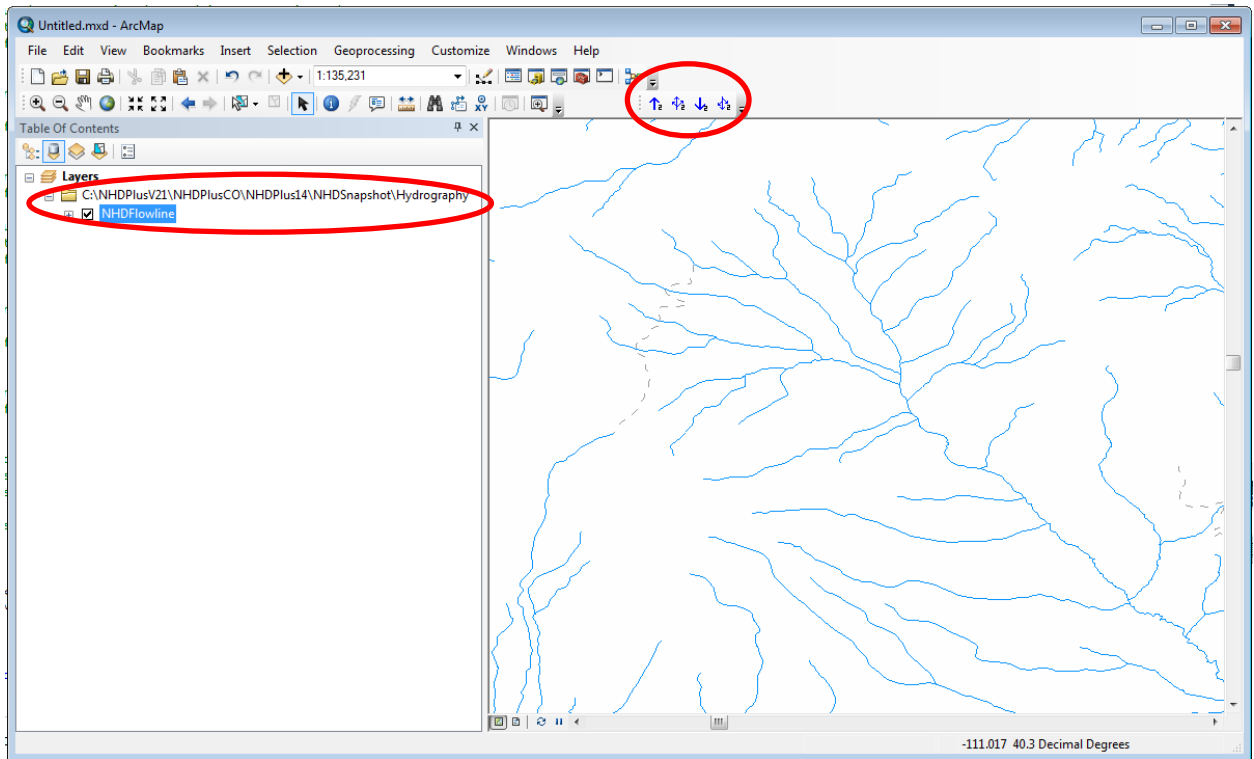
The NHDPlusV2 VAA Nav Toolbar is an ArcMap Toolbar with 4 tools representing the 4 primary navigation types and a form allowing users to identify stop conditions and navigation results filtering rules.

Each navigation involves obtaining information from the user, building a navigation database if one does not exist, clearing any previous navigation results, calling the NHDPlusV2Navigator, exporting the results as an event table dbf, and, finally, rendering the results in the map document using the event table dbf and the active NHDFlowline feature class.

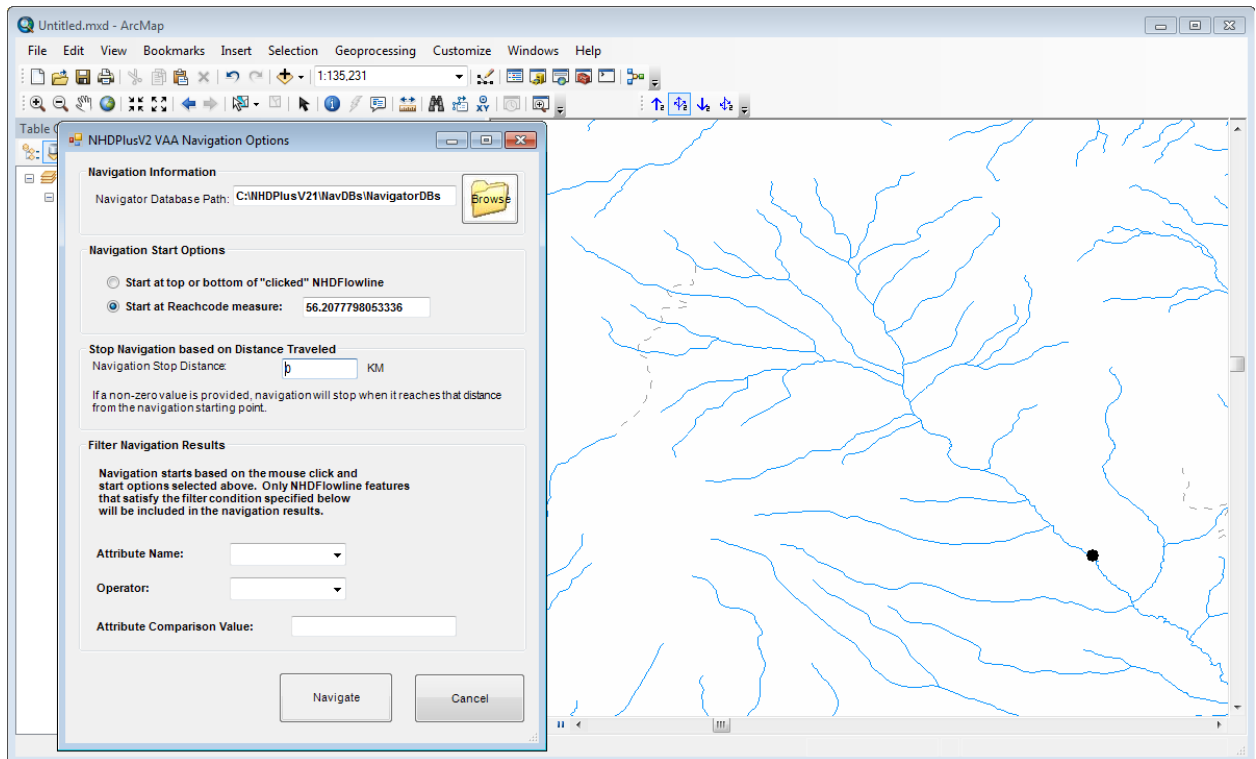
The NHDPlusV2 VAA Nav Toolbar obtains navigation information from the user based on tool selection, a mouse click on a starting comid and measure, and the NHDPlusV2 VAA Navigation Options form.

2. General Usage:

- a. Within ArcMap, manually load a \NHDSnapshot\Hydrography\NHDFlowline shapefile from an NHDPlusV2 workspace.

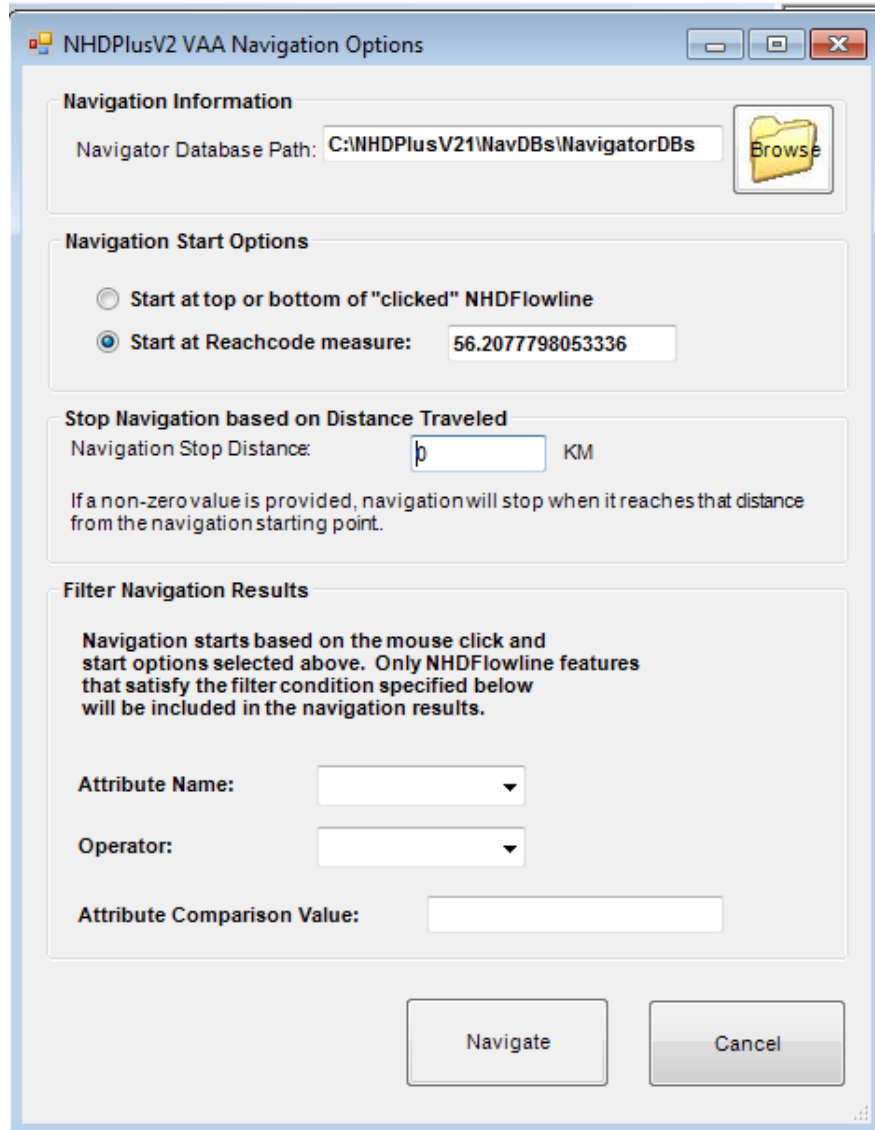


- b. Select the desired tool from the NHDPlusV2 VAA Navigator Toolbar.
(Upstream Mainstem, Upstream with Tributaries, Downstream Mainstem, or Downstream with Divergences)
- c. Click on the desired starting NHDFlowline feature. Zoom in if necessary.



In the above picture, the black dot has been placed at the location of the mouse click and the Navigation Options dialog is open.

- d. The “Navigator Database Path” entry must point to a path where you have read/write privileges. This is the location where the navigator will place the MS SQL database discussed above. This database is created the first time you navigate a specific VPU and then reused each time you perform a navigation in the same VPU. The first navigation in a given VPU will take a little longer in order for the database to be built. The database is named V02NAVDB_<VPUID> (e.g. V02NAVDB_05). Once you have set this path in the dialog, it should appear in the dialog during subsequent navigations.
- e. Establish start, stop and filtering conditions if necessary using the Navigation Options dialog.



NHDPlusV2 VAA Navigation Options Dialog

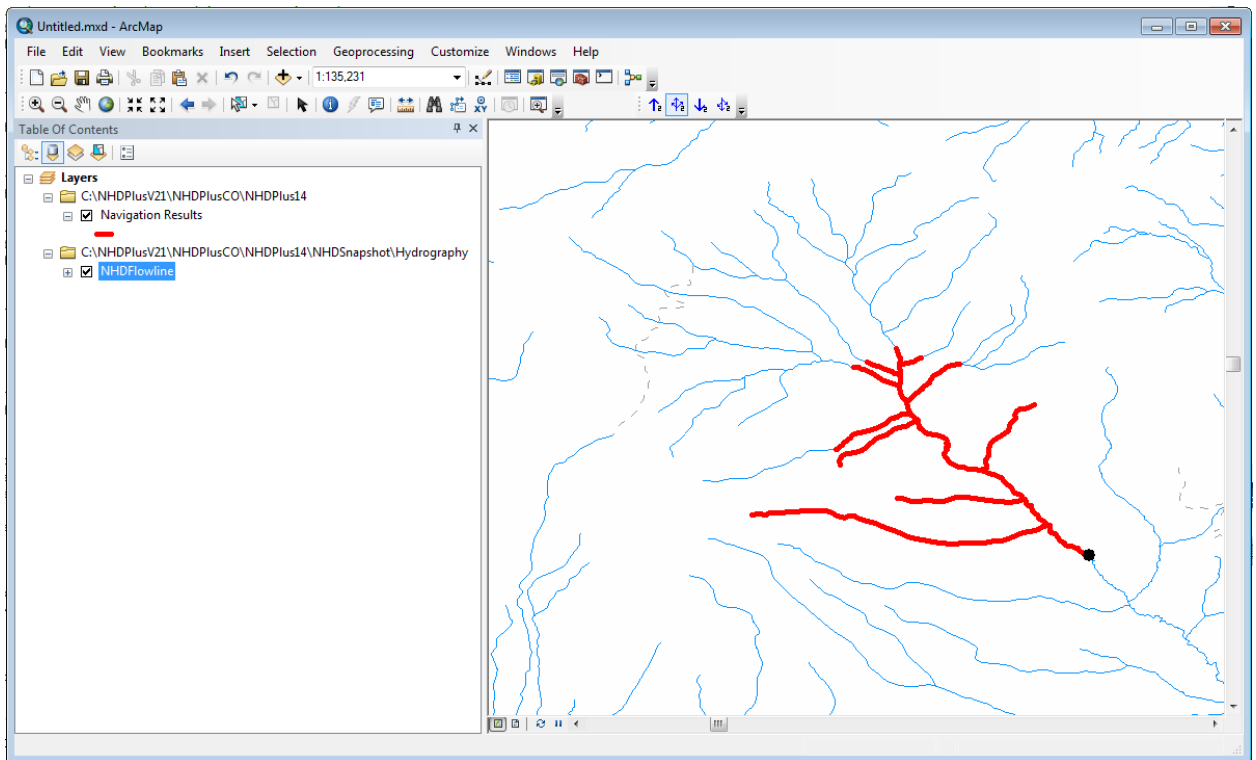
1. Navigation Start Options.

Choose “Start at the top or bottom of the ‘clicked’ NHDFlowline” to include the whole starting NHDFlowline feature in the navigation results. Including the whole starting NHDFlowline feature means that the navigation will begin at the “from” measure (i.e. bottom) of the NHDFlowline feature for upstream navigations and at the “to” measure (i.e. top) of the NHDFlowline feature for downstream navigations.

Choose “Start at Reachcode measure” and provide a measure value to start somewhere else along the NHDFlowline feature. The default value is the measure at the mouse click.

2. Stop navigation based on a stop distance.

Supply a stopping distance other than 0 to stop the navigation when it has traveled the specified distance. 0 indicated that there is no stop distance and the navigation proceeds to the end of the network. The picture below shows an “Upstream with tribs” navigation was stopped after navigating 10 km.



3. Filter navigation results based on a specified value of a selected NHDPlus attribute.

Select an Attribute Name.

The possible NHDPlus attribute names are continuous numeric fields shown in the table below. See the NHDPlusV2 User Guide for additional information about these NHDPlus attributes.

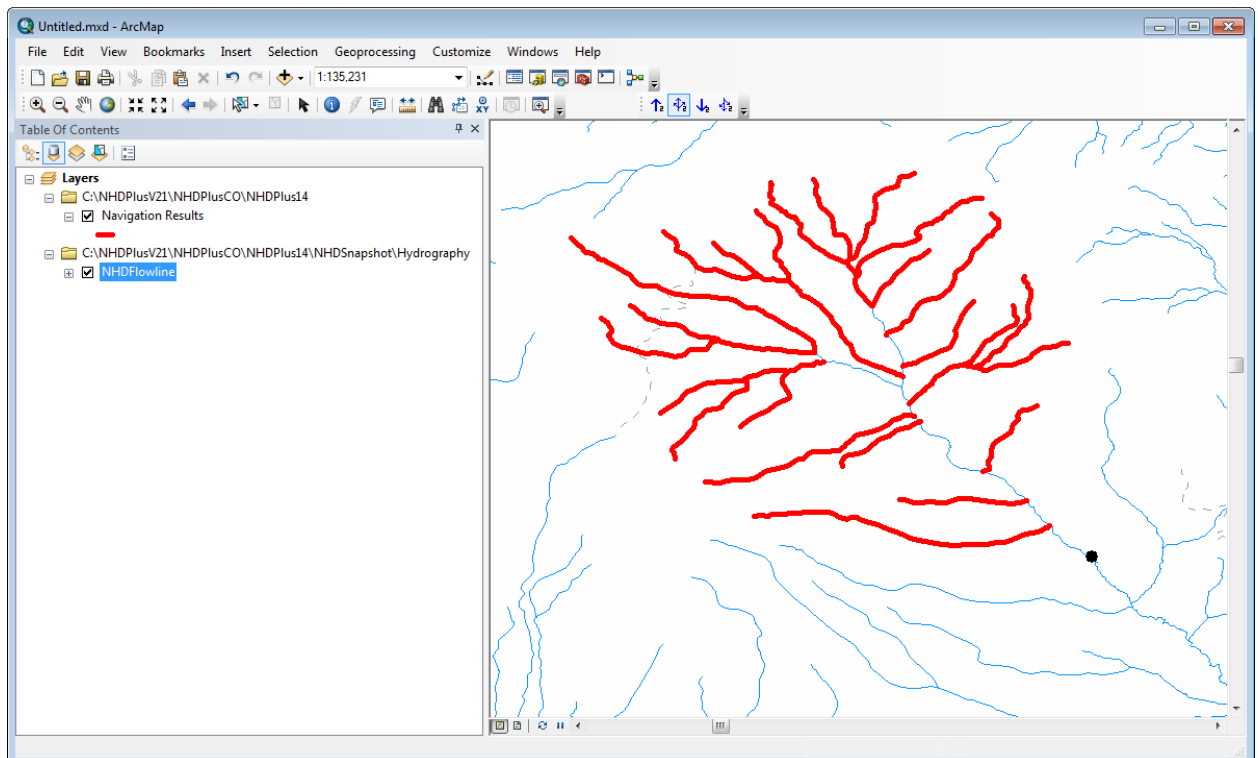
PathLength	Distance to the terminal NHDFlowline feature downstream along the mainpath	Continuous Numeric(13,4)
------------	--	--------------------------

ArbolateSum	Kilometers of stream upstream of the bottom of the NHDFlowline feature	Continuous Numeric(13,4)
TotDASqKm	Total Upstream Cumulative Drainage Area at the downstream end of the NHDFlowline feature	Continuous Numeric(14,6)
DivDASqKm	Divergence-routed Cumulative Drainage Area at the downstream end of the NHDFlowline feature	Continuous Numeric(14,6)

Select an operator

- < Less than
- <= Less than or equal to
- > Greater than
- >= Greater than or equal to

The picture below shows a navigation that was filtered for NHDFlowline features that had TotDASqKM <= 20 sqkm.



- f. Press the “Navigate” button on the Navigation Options dialog to start the navigation. Depending on the options and the size of the navigation, the navigator may take a few moments. The Navigator sets the cursor to a “busy” style, however, there are several parts of the Navigation when ArcMap takes control of the cursor and it may not be displayed as a “busy” style. To ensure that the navigation has enough time to complete, please wait for (1) the “Navigation Results” layer (in red) to display or (2) for a message box to appear that indicates there was an error during navigation.

Using the NHDPlusV2 VAA Navigator from a User-written Program

The NHDPlusV2 VAA Navigator consists of three parts (classes or modules):

LoadSqlServerDB – Loads PlusFlowlineVAA, PlusFlow, and MegaDiv data for a single VPU into the working SQL Server database. This module can be called multiple times to load as many VPUs necessary for the navigations.

MakeWorkingTable – Creates a working table to be used for a single navigation from the data previously loaded into the database via LoadSqlServerDB. The working table is named t<sessionid>_vaa, where <sessionid> is a value generated by the user’s program using system date and time in the format of yyyyymmddhhmmssss. Using a unique sessionid enables the Navigator to be called concurrently by multiple user programs.

V02Navigator – Performs a navigation and places the results in a SQL Server table named t<sessionid>_navresults. This table is overwritten each time the navigator is called with the same <sessionid>.

Properties:

The properties in **RED** are required by the Navigator.

TempWorkAreaPath – input property

Type: String

Applies to calls to: LoadSqlServerDB, MakeWorkingTable

Value: Path to a temporary work area. The user must have read/write privileges to this path. Example: TempWorkAreaPath=”D:\Working”

DataSource – input property

Type: String

Applies to calls to: LoadSqlServerDB, MakeWorkingTable, V02Navigator

Value: Name of the SQL Server datasource on your local computer. Example: Datasource=”(LocalDB)\v11.0”

AddToExisting – input property

Type: Boolean

Applies to calls to: LoadSqlServerDB

Value:

False – for the first VPU

True – for second and subsequent VPUs

DatabaseLocation – input property

Type: String

Applies to calls to: LoadSqlServerDB, MakeWorkingTable, V02Navigator

Value: Path to the navigation SQL Server database. The user must have read/write privileges to this path. Example: DatabaseLocation="D:\NHDPlusV21Data\Navdbs"

DatabaseName – input property

Type: String

Applies to calls to: LoadSqlServerDB, MakeWorkingTable, V02Navigator

Value: Name for the navigation SQL Server database

Recommendation: Use a drainageID (DD) or a VPUid (VVVVVVVV) in the name, as appropriate.

InputNHDPlusLocation – input property

Type: String

Applies to calls to: LoadSqlServerDB

Value: Upper level VPU folder location (i.e. this folder will be named \NHDPlus<DD>\NHDPlus<vvvvvvvv> where DD is the drainage area (e.g. MS) and vvvvvvvv is the vpuid (e.g. 05)

SessionID – input property

Type: String Applies to calls to: MakeWorkingTable, V02Navigator

Value: SessionID for the navigation, unique value based on the computer system date and time allowing multiple concurrent calls to the Navigator

WorkingTableName – input/output property

Type: String

Applies to calls to: MakeWorkingTable (output), V02Navigator (input)

Value: Table name (of the form t<sessionid>_VAA) for the working table that holds the VPU data. This table is overwritten each time MakeWorkingTable is called with the same <sessionid>.

Navtype – input property

Type: String

Applies to calls to: V02Navigator

Value: UPMAIN or UPTRIB or DNMAIN or DNDIV

StartComid – input property

Type: Numeric

Applies to calls to: V02Navigator

Value: Starting NHDFlowline Comid for the navigation, must be greater than 0

StartMeasure – input property

Type: Numeric

Applies to calls to: V02Navigator

Value: Starting measure for the navigation, must be between 0 and 100 inclusive or -1

For -1, the Navigator starts at the “from” measure (i.e. bottom) of the NHDFlowline feature for upstream navigations and at the “to” measure (i.e. top) of the NHDFlowline feature for downstream navigations.

MaxDistance – input property

Type: Numeric

Applies to calls to: V02Navigator

Value: Maximum travel distance in kilometers, greater than or equal to 0

A value of 0 indicates no maximum travel distance will be applied and that the Navigator will navigate each path until the path ends (i.e. at a headwater or a network terminus). 0 is the default value.

ReturnValue – output property

Type: Integer

Applies to calls to: LoadSqlServerDB, MakeWorkingTable, V02Navigator

Value: Return value of a process. 0 for success, 1 otherwise.

ProcessStatus – output property

Type: Integer

Applies to calls to: LoadSqlServerDB, MakeWorkingTable, V02Navigator

Value:

900 – unsuccessful completion

0 - successful completion

ProcessMessage – output property

Type: String

Applies to calls to: LoadSqlServerDB, MakeWorkingTable, V02Navigator

Value: For unsuccessful completions, there is a message about the problem that was encountered.

Methods:

LoadSQLServerDB –

Function: Loads \NHDPPlusAttributes\PlusFlowlinevaa, PlusFlow and MegaDiv for a VPU into the working database.

MakeWorkingTable –

Function: Creates the working table to be used for a navigation from the VPU data loaded into the working database.

VAANavigate –

Function: Performs a navigation. Returns 1 if there is a known problem, 0 otherwise. Results of the navigation are placed in t<sessionid>_NavResults in the working SQLServer database.

VB.NET Example:

The following VB.NET code performs navigation using the NHDPlusV2 VAA Navigator. Add NHDPlusV2 VAA Navigator as a project reference.

```
'Declarations
Dim objLoadDB As NHDPlusV2Navigator.clsLoadSqlServerDB
Dim objMakeWorking As NHDPlusV2Navigator.clsMakeWorkingTable
Dim objNavigate As NHDPlusV2Navigator.clsV02Navigator
Dim intReturn as integer

'Create method object
objLoadDB = New NHDPlusV2Navigator.clsLoadSqlServerDB

'Set SQL properties
objLoadDB.DataSource = "(LocalDB)\v11.0"
objLoadDB.DatabaseLocation = "C:\NHDPlusV2Data\Working"
objLoadDB.DatabaseName = "V02NavDB"
objLoadDB.SQLConnectionTimeout = 120
objLoadDB.SQLCommandTimeout = 120

'Set path to a working folder where processing can occur.
objLoadDB.TempWorkAreaPath = "C:\NHDPlusV2Data\Working"

'The session id is used to prevent conflicts between two similar
'processes running as the same time. Build it from the system
'data/time
objLoadDB.SessionID = "yyyyymmddhhmmssss"

'Call LoadDB for all VPUS needed by your navigations.
'Put this code section in a loop
'Note: The NHDPlusV2 VAA Nav Toolbar always operates only on one VPU.

'Set AddToExisting property equal to true for everything except the
'first VPU.
objLoadDB.AddToExisting = False

'Begin Loop
'Set path to VPU data to be loaded. DD is Drainage Area. VV is VPU id.
objLoadDB.InputNHDPlusLocation = "C:\NHDPlusV2Data\NHDPlusDD\NHDPlusVV"

'Call the method to load the data
```

```

intReturn = objLoadDB.LoadSQLServerDB()
If intReturn <> 0 Then
    strReturn = "Return Value: " + intReturn.ToString + vbCrLf + _
    "ProcessStatus: " + objLoadDB.ProcessStatus.ToString + vbCrLf + _
    "ProcessMessage: " + objLoadDB.ProcessMessage
    Exit Try
End If
AddToExisting = True
'End Loop

```

'For each navigation, call MakeWorking followed by Navigate.

```

'Create MakeWorking method object and properties
objMakeWorking = New NHDPlusV02Navigator.clsMakeWorkingTable
objMakeWorking.DataSource = objLoadDB.DataSource
objMakeWorking.DatabaseLocation = objLoadDB.DatabaseLocation
objMakeWorking.DatabaseName = objLoadDB.DatabaseName
objMakeWorking.SQLConnectionTimeout = 120
objMakeWorking.SQLCommandTimeout = 120
objMakeWorking.TempWorkAreaPath = objLoadDB.TempWorkAreaPath
objMakeWorking.SessionID = objLoadDB.SessionID

```

```

'Create Navigate method object and properties
objNavigate = New NHDPlusV2Navigator.clsV02Navigator
objNavigate.DataSource = objLoadDB.DataSource
objNavigate.DatabaseLocation = obj.LoadDB.DatabaseLocation
objNavigate.DatabaseName = obj.LoadDB.DatabaseName
objNavigate.SessionID = objLoadDB.SessionID
objNavigate.WorkingTableName = objMakeWorking.WorkingTableName
objNavigate.SQLConnectionTimeout = 120
objNavigate.SQLCommandTimeout = 120

```

Put this code block in a loop and execute block for each navigation

```

'Begin Loop
'Set the starting Comid for the navigation
objMakeWorking.StartComid = nnnnnnnn
objMakeWorking.navtype = "UPTRIB"
intReturn = objMakeWorking.MakeWorkingTable()
If intReturn > 0 Then
    strReturn = "Return Value: " + intReturn.ToString + vbCrLf + _
    "ProcessStatus: " + objMakeWorking.ProcessStatus.ToString + vbCrLf + _
    "ProcessMessage: " + objMakeWorking.ProcessMessage
    "Working Table Name: " + objMakeWorking.WorkingTableName
    Exit Try
End If

```

```

objNavigate.StartComid = objMakeWorking.StartComid
'Set the starting measure along the comid where the navigation is to
start. -1 means start at the bottom for upstream navigations and start
at the top for downstream navigation.
objNavigate.StartMeasure = -1
objNavigate.NavType = objMakeWorking.navtype
'Set maximum distance to navigate. 0 means navigate to top of network.
objNavigate.MaxDistance = 0

```

```
'Call navigator
intReturn = objNavigate.VAANavigate()
if intReturn <> 0 then
    strReturn = "Return Value: " + intReturn.ToString + vbCrLf + _
        "ProcessStatus: " + objNavigate.ProcessStatus.ToString + vbCrLf + _
        "ProcessMessage: " + objNavigate.ProcessMessage
End if
'End Loop
```

Python Example:

The following Python script performs navigation using the NHDPlusV2 VAA Navigator. Lines beginning with # are comments. The print statements are for debugging purposes only.

```
import sys, os, time, string, win32com.client, shutil, datetime, arcpy
from arcpy import env

#Initialize objects
o1 = win32com.client.Dispatch("NHDPlusV2Navigator.clsLoadSQLServerDB")
o2 = win32com.client.Dispatch("NHDPlusV2Navigator.clsMakeWorkingTable")
o3 = win32com.client.Dispatch("NHDPlusV2Navigator.clsV02Navigator")

#Set properties and Call LoadSqlServerDB
#Set SQL Server data source
o1.DataSource = "(LocalDB)\v11.0"
#Set location of navigation database
o1.DatabaseLocation = "C:\\NHDPlusV2Data\\Working"
#Set name of navigation database
o1.DatabaseName = "V02NavDB"
o1.SQLConnectionTimeout = 120
o1.SQLCommandTimeout = 120
#Set path to working area
o1.TempWorkAreaPath = "C:\\NHDPlusV2Data\\Working"

#Put this code block in a loop and execute for each VPU needed by your
#navigations.
#Note: The NHDPlus V2 VAA Nav Toolbar always operates only on one VPU.

#Set AddToExisting property equal to true for everything except the
#first VPU.
o1.AddToExisting = False

#Begin Loop
#set path to an NHDPlus VPU. Where DD is Drainage area and VV is VPU id
o1.InputNHDPlusLocation = "C:\\NHDPlusV2\\NHDPlusDD\\NHDPlusVV"

#Call the method to load the data for the VPU
intReturn = o1.LoadSQLServerDB
print "LoadSQLServerDB Return Value: " + str(intReturn)
if intReturn > 0:
    print "Return Value: " + str(intReturn)
    print "ProcessStatus: " + str(o1.ProcessStatus)
    print "ProcessMessage: " + o1.ProcessMessage
o1.AddToExisting = True

#end loop - Do LoadSQLServerDB for each VPU that is needed for the navigation
results

#For each navigation, call MakeWorking followed by Navigate

#Create MakeWorking method object and set properties
```

```

o2.DataSource = o1.DataSource
o2.DatabaseLocation = o1.DatabaseLocation
o2.DatabaseName = o1.DatabaseName
o2.TempWorkAreaPath = o1.TempWorkAreaPath
o2.SessionID = o1.SessionID
o2.SQLConnectionTimeout = 120
o2.SQLCommandTimeout = 120

#Create V02Navigator object and set properties
o3.DataSource = o2.DataSource
o3.DatabaseLocation = o2.DatabaseLocation
o3.DatabaseName = o2.DatabaseName
o3.SessionID = o2.SessionID
o3.SQLConnectionTimeout = 120
o3.SQLCommandTimeout = 120

#Begin Loop

#Set the comid where the navigation is to begin
o2.StartComid = nnnnnnnn
#Set the Navigation type
o2navtype = "UPTRIB"

#Call MakeWorkingTable
intReturn = o2.MakeWorkingTable
print "MakeWorkingTable Return Value: " + str(intReturn)
print "Working Table Name: " + o2.WorkingTableName
if intReturn > 0:
    print "Return Value: " + str(intReturn)
    print "ProcessStatus: " + str(o2.ProcessStatus)
    print "ProcessMessage: " + o2.ProcessMessage

#Navigate
o3.StartComid = nnnnnnnn
o3.StartMeasure = -1
o3.NavType = o2.NavType
o3.MaxDistance = 0
o3.WorkingTableName = o2.WorkingTableName
intReturn = o3.VAANavigate
if intReturn > 0:
    print "Return Value: " + str(intReturn)
    print "ProcessStatus: " + str(o3.ProcessStatus)
    print "ProcessMessage: " + o3.ProcessMessage
else:
    print "Navigation completed successfully. Results are in your SQL Server
database."

#End loop

#Terminate the objects
o1 = None
o2 = None
o3 = None

```