Author: Linda Jun

Created: 2014.06.02

# File Path Handling in GMAT

# Current issues with file path

## GMT-2344 GMAT does not work with file association system

Created: 2011-07-12

Description:

If I create a file with a custom extension (say, .gmat) and associate it with GMAT.exe in Windows, I should be able to double-click the file and have it open in GMAT.

Right now this doesn't work because when GMAT opens, it can't find the startup file. I haven't confirmed, but I'm guessing it's because Windows opens GMAT in the working directory of my file, and GMAT looks in the working directory for gmat\_startup\_file.txt.

This would be extremely useful behavior, and I believe the fix is to make GMAT looks in the same folder as GMAT.exe for the startup file, not in the current working directory.

Status:

I think this was resolved on 2011-12-15 for Windows. A new method FileUtil::GetApplicationPath() was added to get the cross-platform application full path.

This does not work on the Mac. The last time I tried it, the file would associate correctly with GMAT, so that GMAT is opened when the file is double-clicked. But it does not load the script.

## GMT-2688 GMAT should look in the working directory for scripts on command line

Created: 2012-05-15

Description:

This should look for a script in the working directory:

> C:\Users\jjparker\Documents\Software\GMAT\Nightly\bin\GMAT.exe MyScript.script

These should be taken as relative to the working directory:

> C:\Users\jjparker\Documents\Software\GMAT\Nightly\bin\GMAT.exe .\MyScript.script

> C:\Users\jjparker\Documents\Software\GMAT\Nightly\bin\GMAT.exe ..\..\MyScript.script

> C:\Users\jjparker\Documents\Software\GMAT\Nightly\bin\GMAT.exe mysubdir\MyScript.script

Status:

This issue was resolved by Tom on 2013-09-30 but backed out the change by Darrel 2014-05-13 because we thought this might have caused the loading issue of icon and DE file during the 2014a release (GMT-4088). CCB needs to review this again.

## GMT-3630 Running GMAT with relative script path is relative to bin, not working directory

Created: 2013-02-21

Description:

This command should run GMAT with the script c:\workingdir\myscript.script.

c:\workingdir> c:\path\to\bin\gmat.exe myscript.script

Instead it looks for c:\path\to\bin\myscript.script. ".\myscript.script" doesn't work either.

Fixing this would allow users to add GMAT.exe to their system path and run scripts from their working dir.

This works fine:

c:\workingdir> c:\path\to\bin\gmat.exe c:\path\to\myscript.script

The only thing that doesn't work is using a relative path to myscript.script. It's looking for the script relative to GMAT.exe, rather than relative to the working dir.

Status:

Fix someday

## GMT-3950 Allow scripts to reference files relative to the script

Created: 2013-05-23

Description:

This is very important for ease of sharing complex scripts.

Currently, sharing a script that uses any custom data has these steps:

Scenario 1

1. UserA places data in GMAT\data

2. UserA references data by filename only (OrbitSpiceKernelName = 'mykernel.bsp')

3. UserA copies all data files referenced in script to workingdir

4. UserA creates a bundle and sends to UserB

5. UserB unbundles the script

6. UserB copies each data file to his/her own GMAT\data folder

7. UserB runs script

Scenario 2

1. UserA places data alonside script in workingdir

2. UserA references data by full path (OrbitSpiceKernelName = 'C:\path\to\workingdir\mykernel.bsp')

3. UserA creates a bundle and sends to UserB

4. UserB unbundles the script

5. UserB edits the script to change all paths to the new location

6. UserB runs script

By allowing a user to define data relative to the script, sharing a script reduces to:

Scenario 3

1. UserA places data alonside script in workingdir

2. UserA references data by filename only (OrbitSpiceKernelName = 'mykernel.bsp')

3. UserA creates a bundle and sends to UserB

4. UserB unbundles the script

5. UserB runs script

I believe the change is as simple as this pseudo-code:

If path is just a filename

Look in script directory

If not found

Look in default directory (from startup file)

Else

Use provided path

Status:

Fix someday

## GMT-4408 Failed to load icon file and to open DE file in the GUI test system and when loading a script

Created: 2014-02-25

Description:

In the GUI test system, lots of tests failed because of GMAT error loading icon "GMATWin32.ico" and file "leDE1941.405". Randomly it fails to load DE file when loading a script.

Sample error messages from GUI test:

SolarSystem exception: Error loading the SPICE Planetary Ehpemeris (SPK) Kernel "..data/planetary\_ephem/spk/DE421AllPlanets.bsp"

SolarSystem exception: Error opening the DE file "C:\Users\Public\Documents\Jazz\trunk\test\gui\System\SystemTests\Events\_Eclipse\_Heo1\Ex\_Events\_Eclipse\_Heo1\Truth Data/..data/planetary\_ephem/de/leDE1900.421"

Solver subsystem exception: Error opening targeter text file ../output/DifferentialCorrectorDC1.data

Sample error messages when loading a script:

\*\*\* WARNING \*\*\* The icon file "../data/graphics/icons/GMATWin32.ico" does not exist for window 'GmatMdiChildFrame' named 'C:\Users\jjparker\Documents\GMAT\jazz\test\script\input\Commands\FRC-19\_Assignment\scripts\Math\_LHS\_Param\_Array.script', so trying with abs path "C:\Users\jjparker\Documents\GMAT\jazz\test\script\input\Commands\FRC-19\_Assignment\scripts/../data/graphics/icons/GMATWin32.ico"

Error opening the DE file "../data/planetary\_ephem/de/leDE1941.405", so trying with abs path "C:\Users\jjparker\Documents\GMAT\jazz\test\script\input\Commands\FRC-19\_Assignment\scripts/../data/planetary\_ephem/de/leDE1941.405"\*\*\* Error occurred during minimum resource creation.

Message: SolarSystem exception: Error opening the DE file "C:\Users\jjparker\Documents\GMAT\jazz\test\script\input\Commands\FRC-19\_Assignment\scripts/../data/planetary\_ephem/de/leDE1941.405"

Status:

Still open but there is a work-around: Use absolute path for ROOT\_PATH in the gmat\_startup\_file.txt.

# Spec for file path handling in GMAT

*[From: http://en.wikipedia.org/wiki/Path\_%28computing%29]*

***“Path****, the general form of the* [*name*](http://en.wikipedia.org/wiki/Filename) *of a* [*file*](http://en.wikipedia.org/wiki/Computer_file) *or* [*directory*](http://en.wikipedia.org/wiki/Directory_%28file_systems%29)*, specifies a unique location in a* [*file system*](http://en.wikipedia.org/wiki/File_system)*. A path points to a file system location by following the directory tree hierarchy expressed in a* [*string*](http://en.wikipedia.org/wiki/String_%28computer_science%29) *of* [*characters*](http://en.wikipedia.org/wiki/Character_%28computing%29) *in which path components, separated by a delimiting character, represent each directory. The delimiting character is most commonly the* [*slash*](http://en.wikipedia.org/wiki/Slash_%28punctuation%29) *("/"), the* [*backslash*](http://en.wikipedia.org/wiki/Backslash) *character ("\"), or colon (":"), though some* [*operating systems*](http://en.wikipedia.org/wiki/Operating_system) *may use a different* [*delimiter*](http://en.wikipedia.org/wiki/Delimiter)*. Paths are used extensively in* [*computer science*](http://en.wikipedia.org/wiki/Computer_science) *to represent the directory/file relationships common in modern operating systems, and are essential in the construction of* [*Uniform Resource Locators*](http://en.wikipedia.org/wiki/Uniform_Resource_Locator) *(URLs).*

*Systems can use either absolute or relative paths. A* ***full path*** *or* ***absolute path*** *is a path that points to the same location on one file system regardless of the present* [*working directory*](http://en.wikipedia.org/wiki/Working_directory) *or combined paths. It is usually written in reference to a* [*root directory*](http://en.wikipedia.org/wiki/Root_directory)*.*

*A* ***relative path*** *is a path relative to the* [*working directory*](http://en.wikipedia.org/wiki/Working_directory) *of the user or application, so the full absolute path will not have to be given.”*

The purpose of this document is to describe how file path should be handled in GMAT with the following goals:

1. File path and filename handling should be consistent throughout the system
2. Platform dependent file access including file path separator should be isolated to one place
3. Data files and scripts should be easily shared between users on cross-platform
4. File path search order should be intuitive to users

GMAT should be able to handle following three types of filename input from a script on all platforms and show the content in GUI resource panel, resource Show Script panel, and saving to script from the GUI. The current filename content is discussed below for each input format.

1) Filename without a path

ex) aReportFile.Filename = myReport.txt

2) Filename with relative path

ex) aReportFile.Filename = ../myReport.txt

3) Filename with absolute path

ex) aReportFile.Filename = C:/mydir/myReport.txt on windows

ex) aReportFile.Filename = /mydir/myReport.txt on Mac or Linux

## Current implementation

### For showing filename in the GUI resource panel:

For input files:

1. Filename without a path: Shows full path constructed from the ROOT\_PATH and appropriate file type \_PATH specified in the startup file.
2. Filename with relative path: The path is constructed from the ROOT\_PATH and appropriate file type \_PATH specified in the startup file without prepending absolute ROOT\_PATH.
3. Filename with absolute path: Shows full path filename as it appears in the script.

For output files:

1. Filename without a path: Shows filename without a path
2. Filename with relative path: Shows filename with relative path as it appears in the script.
3. Filename with absolute path: Shows full path filename as it appears in the script.

### For showing filename in resource Show Script and writing to Script:

For input files:

1. Filename without a path: Some shows only filename and some shows full path filename. The current behavior is not consistent in GMAT. For example, ODEModel PotentialFile shows 'JGM2.cof' without path, but GroundTrackPlot TextureMap shows with full path.
2. Filename with relative path: Shows filename with relative path as it appears in the script.
3. Filename with absolute path: Show full path filename as it appears in the script.

For output files:

1. Filename without a path: Shows filename without a path
2. Filename with relative path: Shows filename with relative path as it appears in the script.
3. Filename with absolute path: Show full path filename as it appears in the script.

### File search order

Currently it assumes all input files are in the /data directory where GMAT.exe is located. The search order implementation is not isolated in one place. For example, The SolarSystem and SpiceInterface implemented own search order for DE file and Spice Kernel. For GUI icon file, it is implemented in GmatAppData.

### File path/name handling

During the initialization of GMAT, FileManager instance is created from the GmatApp and the following steps happen:

1. FileManager sets current working directory
2. FileManager builds minimal set of default file type path maps and file maps such as paths for data and output and log file.
3. GmatApp creates Moderator instance.
4. GmatApp queries FileManager for full startup file path and passes it to Moderator for reading startup file and other initialization.
5. If Moderator initialization is successful, the GmatApp does some more initialization and then sets the main icon file that is returned from the FileManager.
6. GmatApp sets current directory as working directory
7. GmatApp parses command line options
8. GmatApp creates GmatMainFrame and then do necessary actions based on the command options

During the execution, each resource accessing file queries FileManager for full path for the file type. Usually it calls FileManager::GetPathname() for getting full absolute file path and then constructs full pathname and opens the file. This is done during the resource setting mode or is done only one time at the beginning of execution.

## Proposed implementation

### For showing filename in the GUI resource panel:

For input files: Display paths as they are entered by the user

For output files: Display paths as they are entered by the user

### For showing filename in resource Show Script panel and writing to Script:

For input files:

1. Filename without a path: Show only filename.
2. Filename with relative path: Show filename with relative path as it appears in the script.
3. Filename with absolute path: Show full path filename as it appears in the script.

### File search order

Definition of working directory

*[From: http://en.wikipedia.org/wiki/Working\_directory]*

*“In computing, the working directory of a process is a directory of a hierarchical file system, if any,[1] dynamically associated with each process. When the process refers to a file using a simple file name or relative path (as opposed to a file designated by a full path from a root directory), the reference is interpreted relative to the current working directory of the process. So for example a process with working directory /rabbit-hats that asks to create the file foo.txt will end up creating the file /rabbit-hats/foo.txt.*

*On Windows, the current directory can be retrieved by calling GetCurrentDirectory function.”*

***How to get working (current) directory on Windows:***

*[From: http://msdn.microsoft.com/en-us/library/windows/desktop/aa364934%28v=vs.85%29.aspx]*

*DWORD WINAPI GetCurrentDirectory(\_In\_ DWORD nBufferLength, \_Out\_ LPTSTR lpBuffer);*

***Parameters***

*nBufferLength [in]*

*The length of the buffer for the current directory string, in TCHARs. The buffer length must include room for a terminating null character.*

*lpBuffer [out]*

*A pointer to the buffer that receives the current directory string. This null-terminated string specifies the absolute path to the current directory.*

*To determine the required buffer size, set this parameter to NULL and the nBufferLength parameter to 0.*

***Return value***

*If the function succeeds, the return value specifies the number of characters that are written to the buffer, not including the terminating null character.*

*If the function fails, the return value is zero. To get extended error information, call GetLastError.*

*If the buffer that is pointed to by lpBuffer is not large enough, the return value specifies the required size of the buffer, in characters, including the null-terminating character.*

***Remarks***

*Each process has a single current directory that consists of two parts:*

*A disk designator that is either a drive letter followed by a colon, or a server name followed by a share name (\\servername\sharename)*

*A directory on the disk designator*

*To set the current directory, use the SetCurrentDirectory function.*

*Multithreaded applications and shared library code should not use the GetCurrentDirectory function and should avoid using relative path names. The current directory state written by the SetCurrentDirectory function is stored as a global variable in each process, therefore multithreaded applications cannot reliably use this value without possible data corruption from other threads that may also be reading or setting this value. This limitation also applies to the SetCurrentDirectory and GetFullPathName functions. The exception being when the application is guaranteed to be running in a single thread, for example parsing file names from the command line argument string in the main thread prior to creating any additional threads. Using relative path names in multithreaded applications or shared library code can yield unpredictable results and is not supported.*

***How to set working (current) directory on Windows***

*[From: http://msdn.microsoft.com/en-us/library/windows/desktop/aa365530%28v=vs.85%29.aspx]*

*SetCurrentDirectory function*

*BOOL WINAPI SetCurrentDirectory(\_In\_ LPCTSTR lpPathName);*

***Parameters***

*lpPathName [in]*

*The path to the new current directory. This parameter may specify a relative path or a full path. In either case, the full path of the specified directory is calculated and stored as the current directory.*

*The string must not exceed MAX\_PATH characters, including the terminating null character. The final character before the null character must be a backslash ('\'). If you do not specify the backslash, it will be added for you; therefore, specify MAX\_PATH-2 characters for the path unless you include the trailing backslash, in which case, specify MAX\_PATH-1 characters for the path.*

***Return value***

*If the function succeeds, the return value is nonzero.*

*If the function fails, the return value is zero. To get extended error information, call GetLastError.*

The proposed search order for data files without absolute path is as follows:

For Input:

1. Current GMAT working directory
2. Directory from the startup file in the application directory

For Output:

1. Current GMAT working directory if it has relative path
2. Directory from the startup file in the application directory if no absolute path found or filename has no path
3. Application directory

### Add new methods to FileManager

Add a new method GetGmatWorkingDirectory() to FileManager to separate from GetWorkingDirectory(). Currently GetWorkingDirectory() calls system’s GetCurrentDirectory() which can be changed throughout the GMAT session. GetWorkingDirectory() usually called from MATLAB related resource and commands within GMAT. The new SetGmatWorkingDirectory() will be called when GmatApp starts up to save user’s startup directory for locating scripts and data files.

std::string FileManager::GetGmatWorkingDirectory();

void FileManager::SetGmatWorkingDirectory(const std::string &workDir);

Add a new method FindPath() to FileManager to retrieve path for requested filename that conforms to platform. This function searches path based on the file path search order described above and returns appropriate existing path. If no path found for input, it returns blank.

std::string FileManager::FindPath(const std::string &filename, const FileType type, bool forInput)

{

Change filename to conform to platform

(Windows works for both forward and backward slash, so convert if not on Windows)

If filename has no absolute path

Find the file path according to the search order

If path found

Return the path

Else

If for input

Return blank

Else

Return application output directory

Else

Return platform conforming filename

}

Each resource accessing files calls FileManager::FindPath() to get proper path for the filenames and stores in separate member data. This happens during resource setting mode so that it can be shown in the GUI if we decide to show full path filename in the GUI panel.

**Question**: Do we want to show full path in the GUI panel? In this way users know exact location the file will be loaded or written out. We don’t want to save full path when it is written to file or show in Show Script for making script sharing easier.

**Solution**: Do not show absolute path. Show absolute path as the text edit box hint if possible.

### File path/name handling

During the startup of GMAT, FileManager instance is created from the GmatApp and the steps from current implementation above should happen except the first step. In the FileManager constructor, it gets application directory and sets as bin directory. Another change will be that the Moderator sets GMAT working directory when new script is read, so that new GMAT working directory is used in the search path.

During the resource setting mode or initialization, each resource accessing file queries FileManager for full path for the file type such as OUTPUT\_PATH or DE\_PATH. In current implementation, it calls FileManager::GetPathname() for getting full absolute file path but in the new implementation it calls the new method FileManager::FindPath() to get proper path based on the path search order.

## Table of Sample Default Resource Filename

The following table shows how filename content will be different from the current and new implementation. Basically there will be no changes for output files, but input file content will be changed for some resources.

Root = ROOT\_PATH from the startup file

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Current Implementation | New Implementation |
| Input Files | |  |  |
| Spacecraft Model File | Resource Panel | Root/data/vehicle/models/aura.3ds | aura.3ds |
| Show Script | Root/data/vehicle/models/aura.3ds | aura.3ds |
| Saved Script | Root/data/vehicle/models/aura.3ds | aura.3ds |
| Potential File | Resource Panel | Root/data/gravity/earth/JGM2.cof | JGM2.cof |
| Show Script | JGM2.cof | JGM2.cof |
| Saved Script | JGM2.cof | JGM2.cof |
| Texture Map | Resource Panel | Root/data/graphics/texture/\*Body.jpg | \*Body.jpg |
| Show Script | Root/data/graphics/texture/\*Body.jpg | \*Body.jpg |
| Saved Script | Root/data/graphics/texture/\*Body.jpg | \*Body.jpg |

|  |  |  |  |
| --- | --- | --- | --- |
| MATLAB Function Path | Resource Panel | Blank | Blank |
| Show Script | Not showing | Not showing |
| Saved Script | Not showing | Not showing |

|  |  |  |  |
| --- | --- | --- | --- |
| GMAT Function Path | Resource Panel | NewFunctionName.gmf | NewFunctionName.gmf |
| Show Script | NewFunctionName.gmf | NewFunctionName.gmf |
| Saved Script | NewFunctionName.gmf | NewFunctionName.gmf |
| Output Files | |  |  |
| Report File | Resource Panel | ReportFile1.txt | ReportFile1.txt |
| Show Script | ReportFile1.txt | ReportFile1.txt |
| Saved Script | ReportFile1.txt | ReportFile1.txt |
| Ephemeris File | Resource Panel | EphemerisFile1.data | EphemerisFile1.data |
| Show Script | EphemerisFile1.data | EphemerisFile1.data |
| Saved Script | EphemerisFile1.data | EphemeirsFile1.data |
| DC Report File | Resource Panel | DifferentialCorrectorDC1.data | DifferentialCorrectorDC1.data |
| Show Script | DifferentialCorrectorDC1.data | DifferentialCorrectorDC1.data |
| Saved Script | DifferentialCorrectorDC1.data | DifferentialCorrectorDC1.data |
| Eclipse Locator | Resource Panel | LocatedEvents.txt | LocatedEvents.txt |
| Show Script | LocatedEvents.txt | LocatedEvents.txt |
| Saved Script | LocatedEvents.txt | LocatedEvents.txt |