

Contents

1	Introduction.....	2
2	Resource localization in AutoVue	2
2.1	File lookup	2
2.2	Lookup sequence	2
3	Implementing a sample callback.....	3
3.1	Interface	3
3.2	Writing the callback.....	4
3.2.1	A simple example	4
3.3	Sample Source Code Files	4
4	Testing the Sample DLL:.....	5

1 Introduction

In AutoVue 19.3, integrators are given more flexibility on how to locate external resources. This document describes the implementation of the Custom resource resolution DLL in AutoVue Desktop Edition 19.3.

2 Resource localization in AutoVue

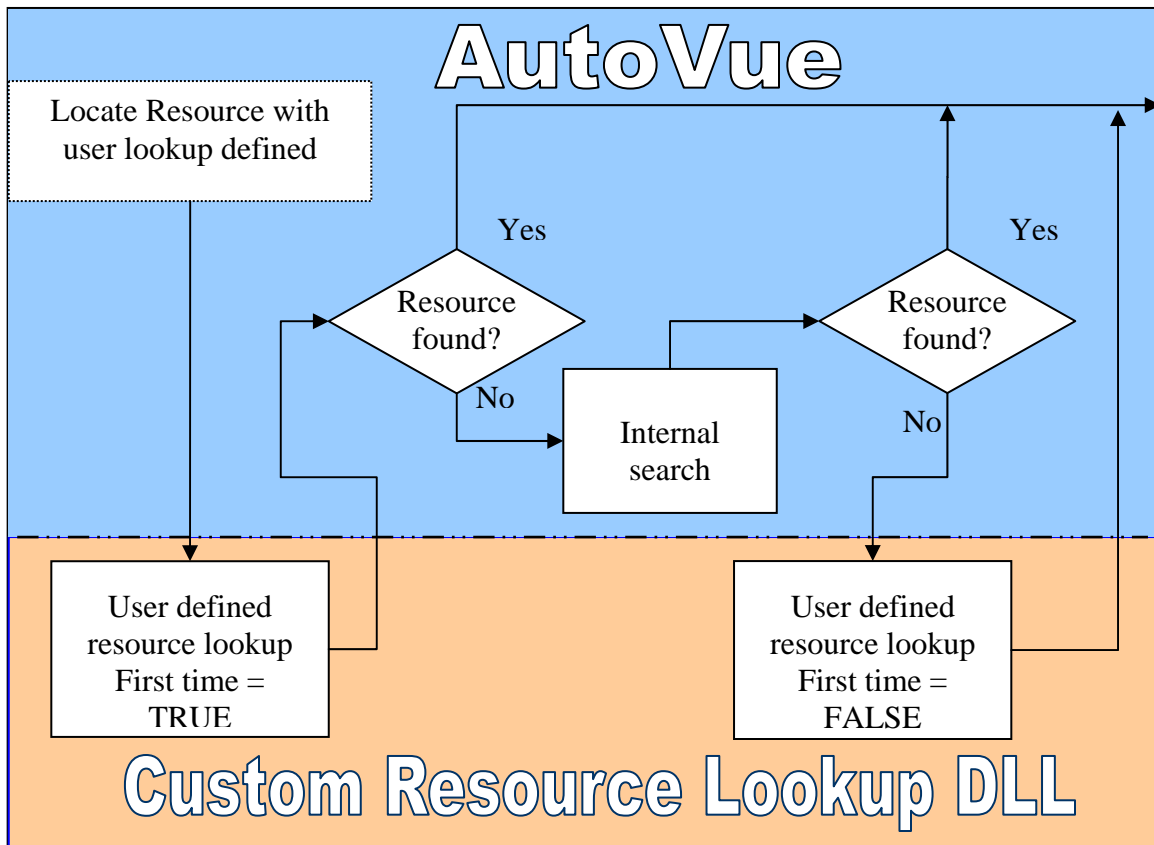
2.1 *File lookup*

As AutoVue is reading a file it might need information that is contained in another file.. This file can be a font file, a line style file, or an external reference (XREF). This process of locating these external files is referred to as resource localization.

The default built-in implementation for resource localization is sufficient in most cases. However, some integrations may require specific knowledge/logic that the default implementation can not handle. 19.3 allows integrators to provide their own custom implementation through a DLL.

2.2 *Lookup sequence*

To locate a resource, AutoVue first checks if there is a user defined DLL and calls it if available. If the DLL call does not locate the resource, AutoVue will use its standard resource localization. If at that point, the resource is still not found, AutoVue will call the user defined DLL again. If that fails, the search is terminated and the resource is flagged as missing. The process is illustrated in the figure below.



3 Implementing a sample callback

3.1 Interface

The callback interface is defined in `UserResLocateDefs.h`. This header file is shipped with AutoVue.

```

/**
 * Function prototype of the DLL callback function
 * @param bPreLocate TRUE if the callback function is called before AutoVue try to locate
the resource
 * @param pResLocateInfo [in] Pointer to structure contains information about the resource
AutoVue try to locate
 * @param pResLocated [out] Pointer to structure contains information about the located
resource
 * @return TRUE if resource is located by the user, FALSE otherwise
 */
typedef BOOL (*DLLLocateProc)(BOOL bPreLocate, const ResLocateInfoStruct*
pResLocateInfo, ResLocatedStruct* pResLocated);
  
```

The custom DLL should contain a C function with the name `UserResLocateProc` that conforms to the prototype shown above.

For AutoVue to locate the custom DLL, it needs to be added to `avwin.ini`:

For a DLL named example.dll that has been copied to the avwin directory, the entry would look like

```
[Options]
RESLOCATEDLL=Example.dll
```

3.2 Writing the callback

3.2.1 A simple example

Below is a sample implementation of a simple search mechanism where a list of mapping for external references is stored in an INI file. The sample lookup DLL will read from the INI file the mapping that matches the resource filename we are trying to locate

Defining the function:

```
/**
 * Implements this callback function to locate resource
 */
extern "C" EXAMPLEDLL_API BOOL UserResLocateProc(BOOL bPreLocate, const
ResLocateInfoStruct* pResLocateInfo, ResLocatedStruct* pResLocated)
{
```

Implementing the search:

```
    wchar_t resolvedPath[_MAX_PATH] = L"";
    if ( GetPrivateProfileStringW(L"Mapping", pResLocateInfo->resPath, L"",
resolvedPath, _MAX_PATH, L"c:/test.map") ) {
```

Returning from a successful lookup:

```
        // Resource located by user
        pResLocated->AddResolvedPathProc(pResLocated->privateContext,
resolvedPath);

        return TRUE;
    }
```

Notifying a failure to locate the file:

```
        // Resource is not located by user, let AutoVue handles it.
        return FALSE;
    }
```

Please refer to the source code for sample custom dll implementation included with AutoVue for more details.

3.3 Sample Source Code Files

This is the list of source files (in the ExampleDLL folder):

- ExampleDLL.cpp
- ExampleDLL.h
- Example.DLL.sln
- Example.DLL.vcproj
- ReadMe.txt
- StdAfx.cpp
- StdAfx.h
- UserLocateDefs.h

These is also a precompiled version of the sample DLL and a map file:

ExampleDLL.dll

test.map

4 Testing the Sample DLL:

Follow steps below to test the implementation using sample *ExampleDLL.dll* and *test.map*:

1. Copy *ExampleDLL.dll* to the avwin directory
2. Copy *test.map* to C:\
3. Copy the base file and its resource files (XREFs or fonts) to separate local folders.
4. Edit *avwin.ini* and set RESLOCATEDLL=ExampleDLL
5. Edit *test.map* file to map the name of the resources to their actual location.
Resource path must match entry in mapping file.
6. Launch AutoVue and view the base file; all the resources that are mapped correctly in *test.map* should be found.