INTEGRATE R INTO YOUR C/C++ APPLICATION

Using Visual C++ 2008

GONG Yu

Outline

- Why We need Integrate R Into Our Application
- Different interface for Integrate R into our application
- How to integrate R into c/c++ app using Visual C++ 2008
- Known issues
- Summary
- Q&A

Why We need Integrate R Into Our Application

- More and more applications need analysis power
- Statistical algorithms are hard to implement
- Time for application develop is so limited
- So why we need write so complicated code in such short time, while they are already implemented by others?

- Don't Reinvent the wheel

Different interface for Integrate R into Our Application

- Interface for C/C++

	Rembedded	RDCOM(used by RExcel)	Pipe(used by Tinn-R)
Core method	Write low-level C/C++ code , linked with R.dll	Using COM tech	Direct use R.exe, through pipe to input data, and receive output
Pros	1.Flexible 2.Performance 3.Low -level	1 .Easy to use 2 .Can used in different language and applications	1.Easy to use 2. Can used in different language
Cons	1.Little Docs 2.Need read R code	1. Need marshaling and un-marshaling 2. Performance 3. Not open source	1. Performance 2. Stability

Different interface for Integrate R into Our Application

- Interface for Other Language

Language	Java	Python
Interface	**rJava**	rpy, rpy2
website	http://www.rforge.net/rJava	rpy.sourceforge.net

Different interface for Integrate R into Our Application

- Those interface(rdcom , rjava , rpy, etc.) all used low-level Rembedded code.
- If we know how to use low-level code to integrate R into application, we can easily develop analysis application base on R.
- How?

Integrate R into C/C++ app using visual C++ 2008

- Prerequisite
  - Rtools(www.murdoch-sutherland.com/**Rtools**)
  - R source code (using subversion)
  - Visual C++ 2008 express (MSDN)
- Procedure
  - Compile R, generate .def file
  - Generate .lib file for visual c++
  - Using visual c++ write our application

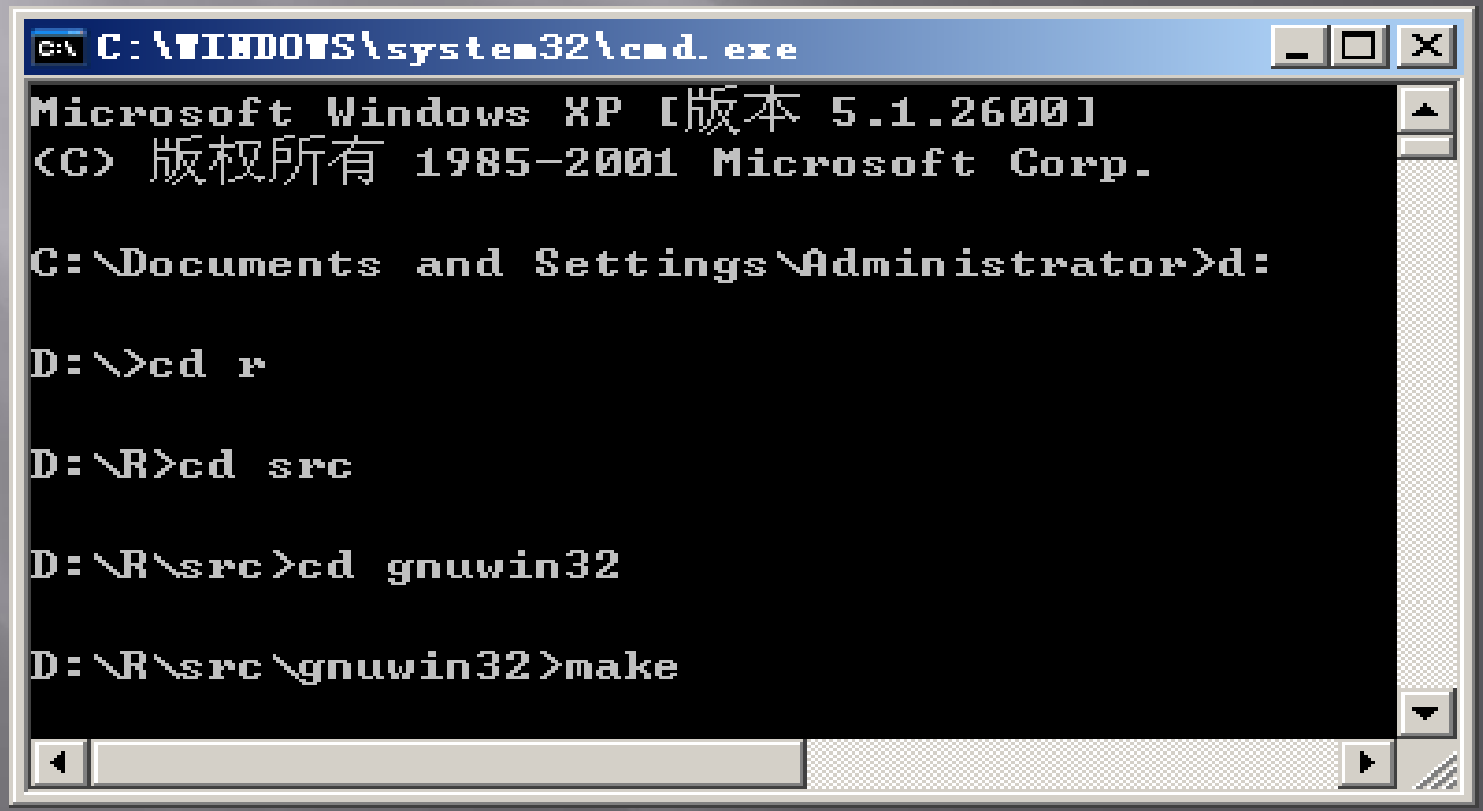
Compile R, generate .def file

Before compile R , Edit src\gnuwin32\makefile

- R.dll: $(OBJS) $(OBJS-EXTRA) $(MAINLIBS) $(EXTRALIBS) dllversion.o
- @$(ECHO) EXPORTS > R.def
- @$(NM) $^ | $(SED) -n 's/^.* [BCDRT] _/ /p' | $(SORT) | uniq > R0.def
- @comm -23 R0.def Rdll.hide >> R.def
- cp R.def ../../R.def
- $(DLL) -shared $(DLLFLAGS) $($*-DLLFLAGS) -o $@ R.def $^ $($*-DLLLIBS) $(DLLLIBS)
- @$(RM) R.def R0.def

compile R code, generate .def file

- Run CMD, enter the src\gnuwin32 dir
- Type make ,then press enter key



Compile R, generate .def file

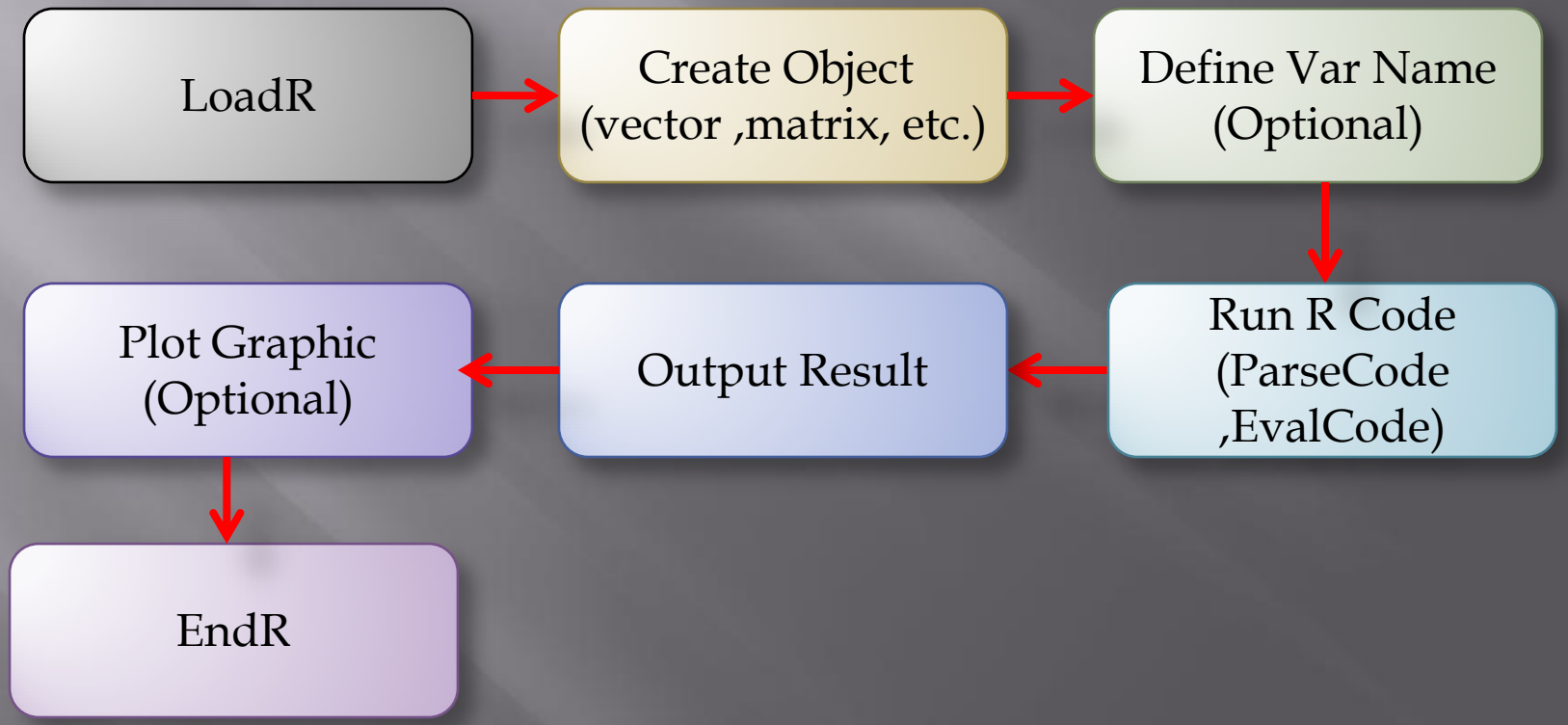
- After compile ,the r.def will in the R root dir
- Or we can use pexports to export the .def file
- http://www.emmestech.com/software/pexports-0.43/download_pexports.html

Generate .lib file for visual c++

- Run Visual Studio 2008 命令提示

using visual c++ write our application

- Use visual c++ 2008 express create an empty console application, then write code



Code Detail

Load R:

```
3 int LoadR(int argc,char **argv){
4  char Rversion[25];
5  sprintf_s(Rversion, "%s.%s", R_MAJOR, R_MINOR);
6  if(strcmp(getDLLVersion(), Rversion) != 0) {
7   return 0;
8  }
9  return Rf_initEmbeddedR(argc,argv);
10 }
```

Code Detail

Create Object : we can create integer, real, string, bool vector or matrix.

Be carful with string encode !!!

```
18  //创建向量并赋值
19  SEXP val1;
20  PROTECT(val1=allocVector(INTSXP, 10));
21  for (int i=0;i<10;i++)
22      INTEGER(val1)[i]=i+1;
23  UNPROTECT(1);
```

Code Detail

Define Var Name :

```
26 int DefineSEXP(char* Name,SEXP val)
27 {
28   ParseStatus error;
29   char* c=Name;
30   SEXP sym=ParseOneLine(c,R_GlobalEnv,&error);
31   if (error!=1)
32   {
33     return 0;
34   }
35
36   if (TYPEOF(sym)==EXPRSXP && LENGTH(sym)>0)
37   {
38     sym=VECTOR_ELT(sym,0);
39   }
40  defineVar((sym)?sym:install(c),val,R_GlobalEnv);
41  return 1;
42 }
```

Code Detail

Run R Code:

```
16 SEXP ParseOneLine(const char *code, SEXP env,ParseStatus *status){
17          SEXP cmd, expr;
18          int errorOccurred=1, retval = 1;
19          PROTECT(cmd = allocVector(STRSXP, 1));
20          SET_STRING_ELT(cmd, 0, mkChar(code));
21          PROTECT(expr = R_ParseVector(cmd, -1, status,R_NilValue));
22          UNPROTECT(2);
23          return expr;
24 }
```

```
44 SEXP ExcuteOneLine(char* cmd,SEXP env,int *errorOccurred)
45 {
46
47   SEXP val,expr;
48   ParseStatus status;
49   expr=ParseOneLine(cmd,env,&status);
50   val=R_tryEval(VECTOR_ELT(expr, 0),env,errorOccurred);
51   return val;
52 }
```

Code Detail

Output Result:

```
32  //自己读取向量的值
33  printf("print value--through our Function\n");
34  for (int i=0;i<LENGTH(val1);i++)
35  {
36   printf("%d value is %d\n",i,INTEGER(val1)[i]);
37  }
```

```c
#include <windows.h>
#include <stdio.h>
#undef ERROR
#include <Rversion.h>
#include <Rembedded.h>
#include <R_ext/RStartup.h>
#include <Rinternals.h>
#include <Rdefines.h>
#include <R_ext/Parse.h>
#include "REngine.h"

int main(int argc,char** argv)
{
 char* c="你好";
 int errorOccurred;
 LoadR(argc,argv);
 //创建向量并赋值
 SEXP val1;
 PROTECT(val1=allocVector(INTSXP, 10));
 for (int i=0;i<10;i++)
    INTEGER(val1)[i]=i+1;
 UNPROTECT(1);
 //定义val的变量名称为“你好”，这一步并不是必需的
 DefineSEXP(c,val1)
 //自己读取向量的值
 printf("print value--through our Function\n");
 for (int i=0;i<LENGTH(val1);i++)
  printf("%d value is %d\n",i,INTEGER(val1)[i]);
 //R图形,使用了 “你好”这个变量
 ExcuteOneLine("windows()",R_GlobalEnv,&errorOccurred);
 ExcuteOneLine("plot(你好)",R_GlobalEnv,&errorOccurred);
 //程序应该最后有这一句，为了显示图形，注释掉了
 //EndR();
 return 0;
}
```

main.c

REngine.h

```
1 #ifndef R_ENGINE_H_
2 #define R_ENGINE_H_
3
4 #define Win32
5 #include <windows.h>
6 #include <stdio.h>
7 #undef ERROR
8 #include <Rversion.h>
9 #include <Rembedded.h>
10 #include <R_ext/RStartup.h>
11 #include <Rinternals.h>
12 #include <Rdefines.h>
13 #include <R_ext/Parse.h>
14 int LoadR(int argc,char **argv);
15 void EndR();
16 SEXP ParseOneLine(const char *code, SEXP env,ParseStatus *status);
17 int DefineSEXP(char* Name,SEXP val);
18 SEXP ExcuteOneLine(char* cmd,SEXP env,int *errorOccurred);
19 #endif /* R_ENGINE_H */
```

```c
#include "REngine.h"
int LoadR(int argc,char **argv){
 char Rversion[25];
 sprintf_s(Rversion, "%s.%s", R_MAJOR, R_MINOR);
 if(strcmp(getDLLVersion(), Rversion) != 0)
    return 0;
 return Rf_initEmbeddedR(argc,argv);
}

void EndR(){Rf_endEmbeddedR(0);}
SEXP ParseOneLine(const char *code, SEXP env,ParseStatus *status){
        SEXP cmd, expr;
        int errorOccurred=1, retval = 1;
        PROTECT(cmd = allocVector(STRSXP, 1));
        SET_STRING_ELT(cmd, 0, mkChar(code));
        PROTECT(expr = R_ParseVector(cmd, -1, status,R_NilValue));
        UNPROTECT(2);
        return expr;
}
int DefineSEXP(char* Name,SEXP val){
  ParseStatus error;
  char* c=Name;
  SEXP sym=ParseOneLine(c,R_GlobalEnv,&error);
  if (error!=1)
    return 0;
  if (TYPEOF(sym)==EXPRSXP && LENGTH(sym)>0)
    sym=VECTOR_ELT(sym,0);
 defineVar((sym)?sym:install(c),val,R_GlobalEnv);
 return 1;
}
SEXP ExcuteOneLine(char* cmd,SEXP env,int *errorOccurred){
 SEXP val,expr;
 ParseStatus status;
 expr=ParseOneLine(cmd,env,&status);
 val=R_tryEval(VECTOR_ELT(expr, 0),env,errorOccurred);
 return val;
}
```

Rengine.c

Known issues

- When Parse R Code(ParseOneLine) we can only know it's right or not, but can't get error message like RGUI or Rconsole does
- When Execute R Code(ExcuteOneLine),we also can't get error message
- To get error message we need modify R source Code(in r-devel mailing list there have some discusses but not correct)

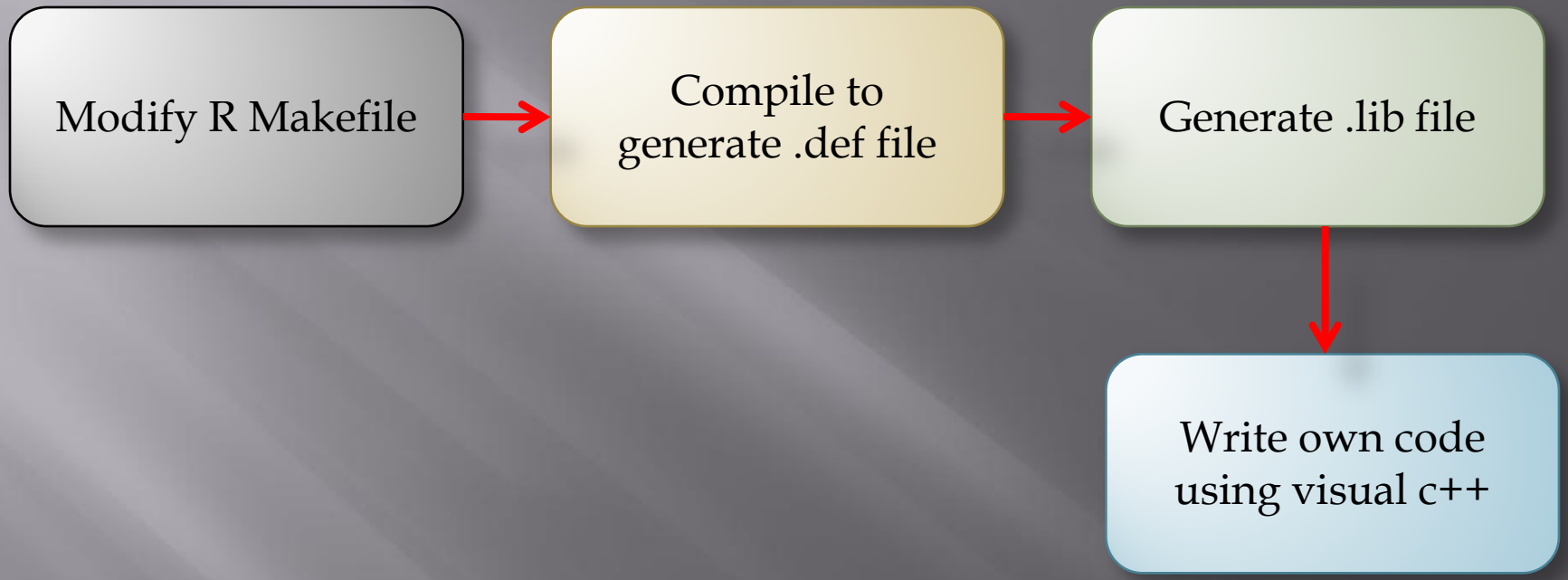
Known issues

- In original PDF output, the Chinese font too ugly
- When need PDF output, use Cairo instead

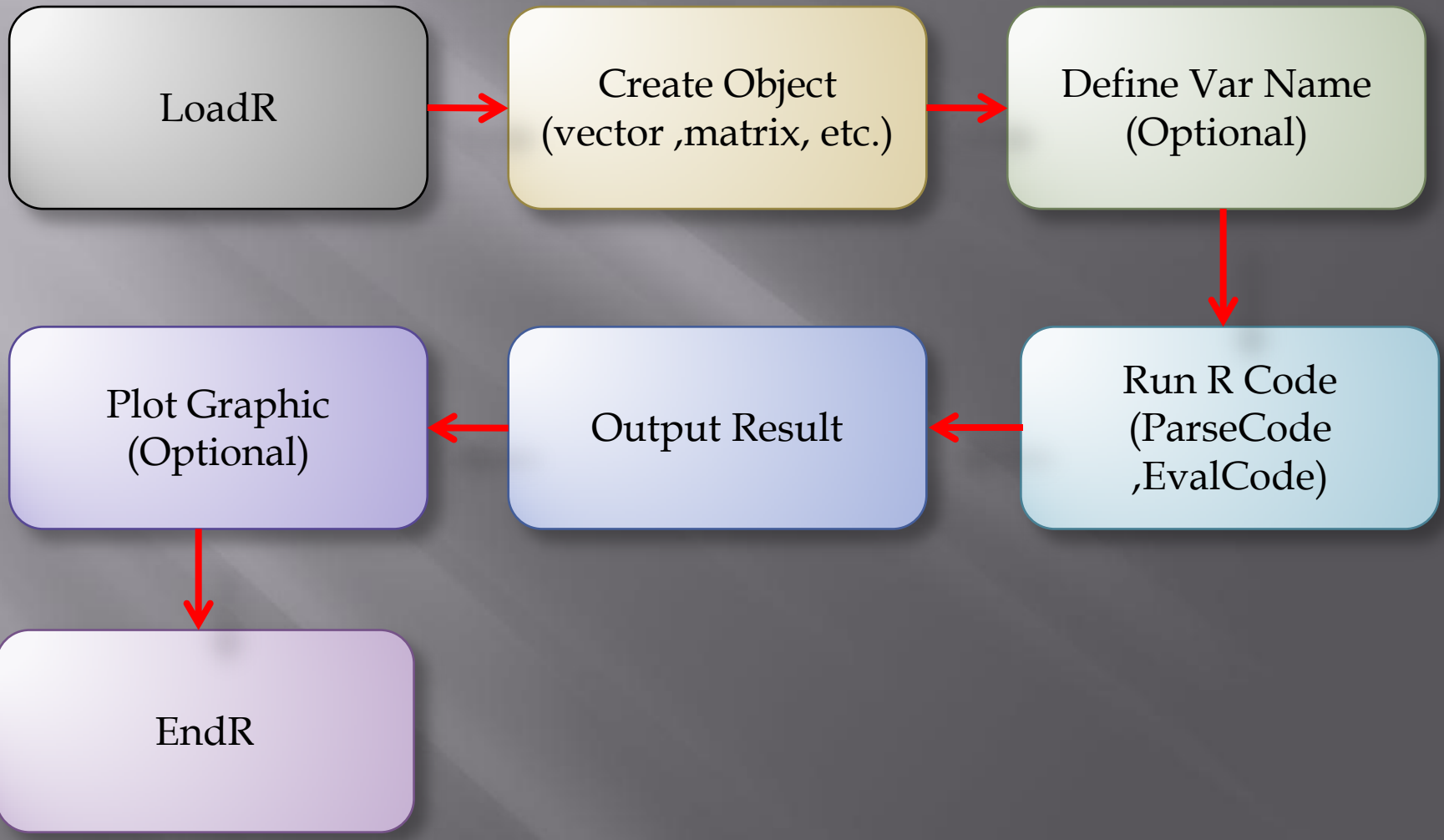
- attention: original Cairo code can't display Chinese character correctly ,need modify, in cairotalk.c change to following !!!

```
#ifdef Win32
 char *Cfontface="Arial Unicode MS";//"Helvetica";
#else
 char *Cfontface="Helvetica";
#endif
```

Summary


Modify R Makefile
Compile to generate .def file
Generate .lib file
Write own code using visual c++

Summary


LoadR
Create Object
(vector ,matrix, etc.)
Define Var Name
(Optional)
Run R Code
(ParseCode
,EvalCode)
Output Result
Plot Graphic
(Optional)
EndR

Summary

- This Slide and sample code only show simplest functions, when you need more advance functions, the best way is reading the R source code(Using Source Insight to view code)

- We also can use visual C++ write a dll ,so it can be used by other language like (delphi ,visual basic ,c# ,etc.)

Reference

- Rserve http://www.rforge.net/Rserve/
- Rjava http://www.rforge.net/rJava/
- R core team .Writing R Extensions

Q&A