

Crystal Reports

Exporting using Print Engine API calls

Overview

This document provides coding examples on how to export at runtime to various formats using Crystal Reports Print Engine (API) Calls. These examples are based on Visual Basic and Visual C++.

Contents

EXPORTING TO AN APPLICATION	2
Visual Basic	2
Visual C++	2
EXPORTING TO DISK.....	3
Visual Basic	3
Visual C++	4
EXPORTING TO MAPI.....	10
Visual Basic	10
Visual C++	10
EXPORTING TO EXCHANGE	11
Visual Basic	11
Visual C++	11
EXPORTING TO ODBC	11
Visual Basic	11
Visual C++	12
EXPORTING TO HTML.....	12
Visual Basic	12
Visual C++	13
EXPORTING TO VIM.....	13
Visual Basic	13
Visual C++	13
EXPORTING TO LOTUS NOTES	14
Visual Basic	14
Visual C++	14
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	15

Exporting to an Application

Visual Basic

Html Type to a application

```
crPEEExportToApp (Job, "c:\htmlsample.htm"+ chr(0), "u2fhtml.dll" + chr(0),
UXFHTML3Type, 0,0,"", "")
```

Paginated TextType to a application

```
crPEEExportToApp (Job, "c:\samplefile.txt"+ chr(0), "u2ftext.dll" + chr(0),
UXFPaginatedTextType, 0,0,"", "")
```

Visual C++

Html Type to an application

```
char errorbuffer[255] = "Error Code: ";
...
HTMLOptions.structSize = sizeof(UXFHTML3Options);
HTMLOptions.fileName = "c:\\htmlsample.htm";
AppOptions.structSize = sizeof(UXDApplicationOptions);
AppOptions.fileName = "samplefile.htm";
ExportOptions.StructSize = sizeof(PEExportOptions);
lstrcpy(ExportOptions.formatDLLName,"u2fhtml.dll");
ExportOptions.formatType = UXFHTML3Type;
ExportOptions.formatOptions = &HTMLOptions;
lstrcpy(ExportOptions.destinationDLLName,"u2dapp.dll");
ExportOptions.destinationType = UXDApplicationType;
ExportOptions.destinationOptions = &AppOptions;
if(!PEEExportTo(myJob,&ExportOptions))
{
    errornum = PEGetErrorCode(myJob);
    MessageBox(hwnd,lstrcat(errorbuffer,itoa(errornum,buffer,10)), "Call to
PEEExportTo failed!",MB_ICONERROR);
}
}
```

Paginated TextType to a application

```

AppOptions.structSize = sizeof(UXDApplicationOptions);

AppOptions.fileName = "samplefile.txt";

PaginatedOptions.structSize = UXFPaginatedTextOptionsSize;

PaginatedOptions.nLinesPerPage = 6;

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;

lstrcpy(ExportOptions.formatDLLName, "u2ftext.dll");

ExportOptions.formatType = UXFPaginatedTextType;

ExportOptions.formatOptions = &PaginatedOptions;

lstrcpy(ExportOptions.destinationDLLName, "u2dapp.dll");

ExportOptions.destinationType = UXDApplicationType;

ExportOptions.destinationOptions = &AppOptions;

if(!PEExportTo(myJob, &ExportOptions))

    { //error handling

        errornum = PEGetErrorCode(myJob);

        MessageBox(hwnd, lstrcat(errorbuffer, itoa(errornum, buffer, 10)), "Call to
        PEExportTo failed!", MB_ICONERROR);

    }

```

Exporting To Disk

Visual Basic

CSV Format

```

crPEExportToDisk(Job, "C:\test1.csv" + chr(0), "u2fsepv.dll" + chr(0) ,
crUXFCharSeparatedType, 1,0, "*" + chr(0), "~" + chr(0))

```

Ascii Text File

```

crPEExportToDisk(Job, "C:\test1.txt" + chr(0), "u2ftext.dll" + chr(0) ,
crUXFTextType, 0,0, ";" , ";")

```

Paginated Text File

```

crPEExportToDisk(Job, "C:\test1.txt" + chr(0), "u2ftext.dll" + chr(0) ,
crUXFPaginatedtextType, 0,0, ";" , ";")

```

Rich Text Format

```
crPEExportToDisk(Job, "C:\test1.rtf" + chr(0), "u2ftrf.dll" + chr(0) ,
crUXFRichTextFormatType, 0,0, """, """)
```

Tab Separated Text

```
crPEExportToDisk(Job, "C:\test1.txt" + chr(0), "u2ftext.dll" + chr(0) ,
crUXFTabbedTextType, 0,0, """, """)
```

Excel

```
crPEExportToDisk(Job, "C:\test1.xls" + chr(0), "u2fxls.dll" + chr(0) ,
crUXFXls5Type, 0,0, """, """)
```

Word Document

```
crPEExportToDisk(Job, "C:\test1.doc" + chr(0), "u2fwordw.dll" + chr(0) ,
crUXFWordWinType, 0,0, """, """)
```

Crystal Report Type

```
crPEExportToDisk(Job, "C:\test1.rpt" + chr(0), "u2fcr.dll" + chr(0) ,
crUXFCrystalReportType, 0,0, """, """)
```

Record Style Options

```
crPEExportToDisk(Job, "C:\test1.rec" + chr(0), "u2frec.dll" + chr(0) ,
crUXFRecordType, 0,0, """, """)
```

Visual C++

CSV Format

```
SeparationOptions.structSize = UXFCharSeparatedOptionsSize;
```

```
SeparationOptions.useReportNumberFormat = TRUE;
```

```
SeparationOptions.useReportDateFormat = FALSE;
```

```
SeparationOptions.stringDelimiter = "*";
```

```
SeparationOptions.fieldDelimiter = "~";
```

```
DiskOptions.structSize = UXDDiskOptionsSize;
```

```
DiskOptions.fileName = "c:\samplefile.csv";
```

```
ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;
```

```
lstrcpy(ExportOptions.formatDLLName, "uxfsepv.dll");
```

```
ExportOptions.formatType = UXFCharSeparatedType;
```

```
ExportOptions.formatOptions = &SeparationOptions;
```

```
lstrcpy(ExportOptions.destinationDLLName, "uxddisk.dll");
```

```
ExportOptions.destinationType = UXDDiskType;

ExportOptions.destinationOptions = &DiskOptions;

//PEGetExportOptions(myJob, &ExportOptions);

if(!PEExportTo(myJob, &ExportOptions))

    { //error handling

        errornum = PEGetErrorCode(myJob);

        MessageBox(hwnd, lstrcat(errorbuffer, itoa(errornum,
buffer, 10)), "Call to PEExportTo failed!", MB_ICONERROR);

    }
```

Ascii Text File

```
DiskOptions.structSize = UXDDiskOptionsSize;

DiskOptions.fileName = "c:\\samplefile.txt";

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;

lstrcpy(ExportOptions.formatDLLName, "u2ftext.dll");

ExportOptions.formatType = UXFTextType;

lstrcpy(ExportOptions.destinationDLLName, "uxddisk.dll");

ExportOptions.destinationType = UXDDiskType;

ExportOptions.destinationOptions = &DiskOptions;

if(!PEExportTo(myJob, &ExportOptions))

    { //error handling

        errornum = PEGetErrorCode(myJob);

        MessageBox(hwnd, lstrcat(errorbuffer, itoa
(errornum, buffer, 10)), "Call to PEExportTo failed!", MB_ICONERROR);

    }
```

Paginated Text File

```
PaginatedOptions.structSize = UXFPaginatedTextOptionsSize;

PaginatedOptions.nLinesPerPage = 6;
```

```
DiskOptions.structSize = UXDDiskOptionsSize;

DiskOptions.fileName = "sampletext.txt";

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;

lstrcpy(ExportOptions.formatDLLName, "u2ftext.dll");

ExportOptions.formatType = UXFPaginatedTextType;

ExportOptions.formatOptions = &PaginatedOptions;

lstrcpy(ExportOptions.destinationDLLName, "uxddisk.dll");

ExportOptions.destinationType = UXDDiskType;

ExportOptions.destinationOptions = &DiskOptions;

if(!PEExportTo(myJob, &ExportOptions))

    { //error handling

        errornum = PEGetErrorCode(myJob);

                                MessageBox(hwnd,lstrcat(errorbuffer,
itoa (errornum,buffer,10)), "Call to PEExportTo failed!",MB_ICONERROR);

    }

}
```

Rich Text Format

```
DiskOptions.structSize = UXDDiskOptionsSize;

DiskOptions.fileName = "c:\\samplefile.rtf";

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;

lstrcpy(ExportOptions.formatDLLName, "u2frtf.dll");

ExportOptions.formatType = UXFRichTextFormatType;

ExportOptions.formatOptions = &SeparationOptions;

lstrcpy(ExportOptions.destinationDLLName, "uxddisk.dll");

ExportOptions.destinationType = UXDDiskType;

ExportOptions.destinationOptions = &DiskOptions;

if(!PEExportTo(myJob, &ExportOptions))

    { //error handling
```

```
        errornum = PEGetErrorCode(myJob);

        MessageBox(hwnd,lstrcat(errorbuffer,
itoa(errornum,buffer,10)), "Call to PEEExportTo failed!",MB_ICONERROR);
    }
}
```

Tab Separated Text

```
DiskOptions.structSize = UXDDiskOptionsSize;
DiskOptions.fileName = "c:\\samplefile.tx2";
ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;
lstrcpy(ExportOptions.formatDLLName, "u2ftext.dll");
ExportOptions.formatType = UXFTabbedTextType;
lstrcpy(ExportOptions.destinationDLLName, "uxddisk.dll");
ExportOptions.destinationType = UXDDiskType;
ExportOptions.destinationOptions = &DiskOptions;
if(!PEExportTo(myJob, &ExportOptions))
    {
        //error handling
        errornum = PEGetErrorCode(myJob);
        MessageBox(hwnd,lstrcat(errorbuffer,
itoa(errornum,buffer,10)), "Call to PEEExportTo failed!",MB_ICONERROR);
    }
}
```

Excel

```
DiskOptions.structSize = UXDDiskOptionsSize;
DiskOptions.fileName = "c:\\testfile.xls";
ExportOptions.StructSize = sizeof(PEExportOptions);
lstrcpy(ExportOptions.formatDLLName,"u2fxls.dll");
ExportOptions.formatType = UXFXls5Type;
lstrcpy(ExportOptions.destinationDLLName,"u2ddisk.dll");
ExportOptions.destinationType = UXDDiskType;
```

```
ExportOptions.destinationOptions = &DiskOptions;
if(!PEExportTo(myJob,&ExportOptions))
{
    errornum = PEGetErrorCode(myJob);
    MessageBox(hwnd,lstrcat(errorbuffer,
itoa(errornum,buffer,10)),"Call to PEExportTo failed!",MB_ICONERROR);
}
```

Word Document

```
DiskOptions.structSize = UXDDiskOptionsSize;
DiskOptions.fileName = "c:\\testfile.doc";
ExportOptions.StructSize = sizeof(PEExportOptions);
lstrcpy(ExportOptions.formatDLLName,"u2fwordw.dll");
ExportOptions.formatType = UXFWordWinType;
lstrcpy(ExportOptions.destinationDLLName,"u2ddisk.dll");
ExportOptions.destinationType = UXDDiskType;
ExportOptions.destinationOptions = &DiskOptions;
if(!PEExportTo(myJob,&ExportOptions))
{
    errornum = PEGetErrorCode(myJob);
    MessageBox(hwnd,lstrcat(errorbuffer,
itoa(errornum,buffer,10)),"Call to PEExportTo failed!",MB_ICONERROR);
}
```

Crystal Report Type

```
DiskOptions.fileName = "c:\\testfile.rpt";
ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;
lstrcpy(ExportOptions.formatDLLName,"u2fcr.dll");
ExportOptions.formatType = UXFCrystalReportType;
lstrcpy(ExportOptions.destinationDLLName,"u2ddisk.dll");
```



```
ExportOptions.destinationType = UXDDiskType;

ExportOptions.destinationOptions = &DiskOptions;

if(!PEExportTo(myJob,&ExportOptions))
{
    errornum = PEGetErrorCode(myJob);

    MessageBox(hwnd,lstrcat(errorbuffer,
itoa(errornum,buffer,10)),"Call to PEExportTo failed!",MB_ICONERROR);
}
```

Record Style Options

```
RecordOptions.structSize = sizeof(UXFRecordStyleOptions);

RecordOptions.useReportNumberFormat = TRUE;

RecordOptions.useReportDateFormat = TRUE;

DiskOptions.structSize = UXDDiskOptionsSize;

DiskOptions.fileName = "c:\\testfile.rec";

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;

lstrcpy(ExportOptions.formatDLLName,"u2frec.dll");

ExportOptions.formatType = UXFRecordType;

ExportOptions.formatOptions = &RecordOptions;

lstrcpy(ExportOptions.destinationDLLName,"u2ddisk.dll");

ExportOptions.destinationType = UXDDiskType;

ExportOptions.destinationOptions = &DiskOptions;

if(!PEExportTo(myJob,&ExportOptions))
{
    errornum = PEGetErrorCode(myJob);

    MessageBox(hwnd,lstrcat(errorbuffer,
itoa(errornum,buffer,10)),"Call to PEExportTo failed!",MB_ICONERROR);
}
```

Exporting To MAPI

Visual Basic

```
crPEExportToMapi (Job, "techsupport@seagatesoftware.com"+ chr(0), "" +  
chr(0), "Test", "Test Worked!", "u2fhtml.dll", UXFHTML3Type, 0,0,"", "")
```

Visual C++

```
MailOptions.structSize = UXDMAPIOptionsSize;  
  
MailOptions.toList = " techsupport@seagatesoftware.com ";  
  
MailOptions.ccList = "";  
  
MailOptions.subject="Test";  
  
MailOptions.message="Test Worked!";  
  
MailOptions.nRecipients = 0;  
  
MailOptions.recipients = 0;  
  
HTMLOptions.structSize = sizeof(UXFHTML3Options);  
  
HTMLOptions.fileName = "htmlsample.htm";  
  
ExportOptions.StructSize = sizeof(PEExportOptions);  
  
lstrcpy(ExportOptions.formatDLLName,"u2fhtml.dll");  
  
ExportOptions.formatType = UXFHTML3Type;  
  
ExportOptions.formatOptions = &HTMLOptions;  
  
lstrcpy(ExportOptions.destinationDLLName,"u2dmapi.dll");  
  
ExportOptions.destinationType = UXDMAPIType;  
  
ExportOptions.destinationOptions = &MailOptions;  
  
if(!PEExportTo(myJob,&ExportOptions))  
{  
  
errornum = PEGetErrorCode(myJob);  
  
MessageBox(hwnd,lstcat(errorbuffer,itoa(errornum,buffer,10)),"Call to  
PEExportTo failed!",MB_ICONERROR);  
  
}
```

Exporting To Exchange

Visual Basic

```
crPEExportToExch (Job, "Seagate's E-mail" + chr(0), "" + chr(0), "Mailbox –
Sample Mailbox@Inbox" + chr(0), "u2fcr.dll" + chr(0),
UXFCrystalReportsType, 0, 0, "", "")
```

Visual C++

```
PostFolderOptions.structSize = UXDPostFolderOptionsSize;

PostFolderOptions.pszProfile      = "Seagate's E-mail";

PostFolderOptions.pszPassword    = "";

PostFolderOptions.wDestType = UXDPostDocMessage;

// pszFolderPath has to be in the following format:
// <Message Store Name>@<Folder Name>@<Folder Name>

PostFolderOptions.pszFolderPath = "Mailbox – Sample Mailbox@Inbox";

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;

lstrcpy(ExportOptions.formatDLLName,"u2fcr.dll");

ExportOptions.formatType = UXFCrystalReportType;

lstrcpy(ExportOptions.destinationDLLName,"u2dpost.dll");

ExportOptions.destinationType = UXDExchFolderType;

ExportOptions.destinationOptions = &PostFolderOptions;

if(!PEExportTo(myJob,&ExportOptions))
{
    errornum = PEGetErrorCode(myJob);

    MessageBox(hwnd,lstrcat(errorbuffer,

itoa(errornum,buffer,10)),"Call to PEExportTo failed!",MB_ICONERROR);

}
```

Exporting To ODBC

Visual Basic

```
crPEExportToODBC (Job, "ODBCExportTest" + chr(0), "" + chr(0), "" +
chr(0), "SampleTable" + chr(0))
```

Visual C++

Export To ODBC - Access

```

ODBCOptions.structSize = sizeof(UXFODBCOptions);

ODBCOptions.dataSourceName =      "ODBCExportTest";
ODBCOptions.dataSourceUserID =     "";
ODBCOptions.dataSourcePassword =   "";
ODBCOptions.exportTableName =      "SampleTable";

DiskOptions.structSize = UXDDiskOptionsSize;
DiskOptions.fileName = "c:\xtreme.mdb";

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;
lstrcpy(ExportOptions.formatDLLName, "u2fodbc.dll");
ExportOptions.formatType = UXFODBCType;
ExportOptions.formatOptions = &ODBCOptions;
lstrcpy(ExportOptions.destinationDLLName, "u2ddisk.dll");
ExportOptions.destinationType = UXDDiskType;
ExportOptions.destinationOptions = &DiskOptions;

if(!PEExportTo(myJob, &ExportOptions))
{
    //error handling

        errornum = PEGetErrorCode(myJob);

        MessageBox(hwnd,lstreat(errorbuffer,

itoa(errornum,buffer,10)), "Call to PEExportTo failed!",MB_ICONERROR);

    }

else

    MessageBox(hwnd,"IT WORKED", "Call to PEExportTo was
successful",MB_OK);

```

Exporting To HTML

Visual Basic

```
crPEExportToHTML (Job, UXFHTML3Type, "c:\newfile.htm" + chr(0))
```

Visual C++

```

HTMLOptions.structSize = sizeof(UXFHTML3Options);

HTMLOptions.fileName = "c:\\htmlsample.htm";

DiskOptions.structSize = UXDDiskOptionsSize;

DiskOptions.fileName = "c:\\newfile.htm";

ExportOptions.StructSize = sizeof(PEExportOptions);

lstrcpy(ExportOptions.formatDLLName,"u2fhtml.dll");

ExportOptions.formatType = UXFHTML3Type;

ExportOptions.formatOptions = &HTMLOptions;

lstrcpy(ExportOptions.destinationDLLName,"u2ddisk.dll");

ExportOptions.destinationType = UXDDiskType;

ExportOptions.destinationOptions = &DiskOptions;

if(!PEExportTo(myJob,&ExportOptions))
    {
        errornum = PEGetErrorCode(myJob);

        MessageBox(hwnd,lstrcat(errorbuffer,

itoa(errornum,buffer,10)),"Call to PEExportTo failed!",MB_ICONERROR);

    }

```

Exporting To VIM**Visual Basic**

```

crPEExportToVIM (Job, "Sample Mailbox" + chr(0), "" + chr(0), "" + chr(0),
"Test" + chr(0), "Test Worked" + chr(0), "u2fcr.dll" + chr(0),
UXFCrystalReportType, 0,0,"","")

```

Visual C++

```

VIMOptions.structSize = UXDVIMOptionsSize;

VIMOptions.toList      = "Sample Mailbox";

VIMOptions.ccList      = "";

VIMOptions.bccList     = "";

VIMOptions.subject     = "Test";

```

```
VIMOptions.message = "The test worked";

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;

lstrcpy(ExportOptions.formatDLLName,"u2fcr.dll");

ExportOptions.formatType = UXFCrystalReportType;

lstrcpy(ExportOptions.destinationDLLName,"u2dvim.dll");

ExportOptions.destinationType = UXDVIMType;

ExportOptions.destinationOptions = &VIMOptions;

if(!PEExportTo(myJob,&ExportOptions))
{
    errornum = PEGetErrorCode(myJob);

    MessageBox(hwnd,lstrcat(errorbuffer,itoa
(errornum,buffer,10)),"Call to PEExportTo failed!",MB_ICONERROR);
}
}
```

Exporting To Lotus Notes

Visual Basic

```
crPEExportToNotes(job, "c:\notes\data\journal.nsf" + Chr(0), "Report Form" +
Chr(0), "Notes Test" + Chr(0), "u2fcr.dll" + Chr(0), UXFCrystalReportType, 0,
0, "", "")
```

Visual C++

```
NotesOptions.structSize = UXDNotesOptionsSize;

NotesOptions.szDBName = "c:\notes\data\journal.nsf";

NotesOptions.szFormName = "Report Form";

// should be "Report Form"

NotesOptions.szComments = "Notes Test";

ExportOptions.StructSize = PE_SIZEOF_EXPORT_OPTIONS;

lstrcpy(ExportOptions.formatDLLName,"u2fcr.dll");

ExportOptions.formatType = UXFCrystalReportType;

lstrcpy(ExportOptions.destinationDLLName,"u2dnotes.dll");

ExportOptions.destinationType = UXDNotesType;
```

```
ExportOptions.destinationOptions = &NotesOptions;
if(!PEExportTo(myJob,&ExportOptions))
{
    errornum = PEGetErrorCode(myJob);
    MessageBox(hwnd,lstrcat(errorbuffer,itoa
(errornum,buffer,10)),"Call to PEExportTo failed!",MB_ICONERROR);
}
```

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

<http://support.crystaldecisions.com/>

Email Support:

<http://support.crystaldecisions.com/support/answers.asp>

Telephone Support:

<http://www.crystaldecisions.com/contact/support.asp>