



# ***CoolSpools User Guide***

*Version 2.1.4  
July 2002*

*@riadne software ltd, cheltenham, england*

# Table of Contents

<b>Introduction .....</b>	<b>5</b>
<b>Applications .....</b>	<b>5</b>
Automated distribution of reports and documents .....	5
Information sharing .....	5
Report Enhancement .....	5
Archiving and offline storage .....	6
<b>Document Formats .....</b>	<b>6</b>
Adobe® PDF (Portable Document Format) .....	6
RTF (Rich Text Format ) .....	7
HTML (Hypertext Mark-up Language) .....	7
Excel Format .....	7
Delimited ASCII Text .....	8
Plain ASCII Text .....	8
Archive format .....	8
<b>Features .....</b>	<b>8</b>
Support for AFP, IPDS and SCS spooled files .....	9
Colorize your reports! .....	9
User-definable Font Mapping .....	9
PDF Bookmarks .....	9
PDF Data Compression .....	10
Web-optimized PDF .....	10
Encrypted and password-protected PDF .....	10
Spooled File Splitting .....	10
User-definable Exit Program .....	10
<b>Installation .....</b>	<b>11</b>
<b>Method 1: Installing CoolSpools using FTP .....</b>	<b>11</b>
<b>Method 2: Installing CoolSpools using the IFS .....</b>	<b>13</b>
<b>Maintenance .....</b>	<b>17</b>
<b>Method 1: Installing a PTF using FTP .....</b>	<b>17</b>

<b>Method 2: Installing a PTF using the IFS .....</b>	<b>19</b>
<b>Using CVTSPLSTMF .....</b>	<b>22</b>
Getting Started.....	22
Command Parameters.....	22
FROMFILE.....	22
TOSTMF .....	23
JOB.....	23
SPLNBR.....	24
TOFMT.....	25
STMFOPT.....	26
TITLE .....	27
PAGESIZE .....	27
CUSTOMPAGE .....	29
MARGINS .....	29
RTFMARGINS .....	30
PASSWORD.....	31
FONT .....	32
INCLFILE .....	34
COLOR .....	36
DFTFONT .....	37
CVTFONTID .....	39
CVTFNTRSC .....	40
BLANKS.....	41
BOOKMARK .....	41
BMARKPOS.....	42
BMARKKEY .....	43
SPLIT.....	45
SPLITPAGE.....	46
SPLITKEY.....	46
EXITPGM.....	47
EXITPGMPRM.....	48
EXITPGMPOS .....	48

EXITPGMKEY.....	49
DELIMITERS .....	50
PDFVIEWER.....	52
EXCLLINBR .....	54
EXCLLINKEY.....	54
WORKSIZE.....	54
TODIR.....	55
STMFCODPAG.....	57
AUT.....	58
<b>Using CVTSTMSPLF .....</b>	<b>60</b>
FROMSTMF.....	60
FROMDIR .....	60
NEWOWN.....	61
OUTPTY .....	61
<b>Appendix A: Sample Exit Program .....</b>	<b>62</b>

## **Introduction**

CoolSpools is a rich set of software tools which enable businesses to adapt their traditional AS/400 reports to the challenges of the new age of e-commerce and the Internet.

Let's have a look at what **CoolSpools** can do for your business.

## **Applications**

The **CoolSpools** product revolves around the **CVTSPLSTMF** (Convert Spooled File to Stream File) command, which takes almost any AS/400 spooled file and converts it into a PC-style document. This popular utility combines ease of use with a wide range of options and features. CoolSpools can help streamline and modernize many areas of your business **without forcing you to re-program** existing applications.

### ***Automated distribution of reports and documents***

Still sending reports out on paper? Maybe you take orders over the Web but still have to send invoices by snail-mail because the invoicing application is an old AS/400 package?

Files created by **CoolSpools** can be e-mailed to users, colleagues and customers, rather than sent out on paper. Not only does this save you money and simplify your business processes, it also makes clear to everyone that your business is responding to the challenges of the Internet era.

### ***Information sharing***

However many copies of a report you print, it's never enough. There's always someone else who'd like to see it but is in an office on the other side of the country and can't get access to a copy.

Files created by **CoolSpools** can be stored on a central corporate server, such as your AS/400, or a PC server, and shared amongst all your users as a corporate information resource.

Alternatively, you could publish them on your Web site for customers worldwide to see, or on a secure Intranet or Extranet.

### ***Report Enhancement***

**CoolSpools** doesn't just convert your reports, it enhances them too. For example, when converting to PDF format, you can add bookmarks that index the pages of your report so your users can find the information they want quickly and easily. When you create a PDF or HTML document from a spooled file, you can add color to improve the presentation as well. Maybe you couldn't afford an

AS/400 color printer: now that's no longer an issue since you can simply convert your spooled file to PDF and print your document on an inexpensive PC color printer!

### ***Archiving and offline storage***

The AS/400 provides no built in means of saving and restoring spooled files. Yet for many companies their AS/400 reports are a critical part of the business process, and may not be easy to re-create. Maybe you have to waste large amounts of expensive AS/400 disk just keeping copies of old reports online. Doing this can also impact your system's performance, since jobs which created spooled files remain in the system even when they have ended if they created spooled files which still exist on an output queue,

Files created by ***CoolSpools*** can be stored on inexpensive storage media such as PC disk or CD-ROM for easy retrieval. Once ***CVTSPLSTMF*** has converted your report into a stream file, you can move this file to a PC server, or save it to tape or to CD-ROM, then delete the original spooled file, freeing up precious AS/400 resources.

***CoolSpools*** offers several options for archiving spooled files. You can convert the spooled file to a PDF file, in which case you would use Adobe Acrobat reader to re-print the report; you can convert it to an RTF (Rich Text Format) file, in which case you would use a Word Processor such as MS Word or Lotus WordPro; or you can use the TOFMT(\*SAV) option, in which case the ***CVTSTMSPLF*** (Convert Stream File to Spooled File) command can be used to restore the original spooled file from the stream file archive.

Both TOFMT(\*PDF) and TOFMT(\*SAV) use data compression to minimize the size of the archived spooled file. The data compression algorithm used is an AS/400 implementation of the industry-standard zlib compression library, developed by Jean-loup Gailly and Mark Adler, to whom our thanks are due.

### ***Document Formats***

Let's have a closer look at how ***CoolSpools*** achieves all this.

***CVTSPLSTMF*** takes an AS/400 spooled file and converts it to a PC document stored as a stream file on the AS/400. The stream file can be saved anywhere in the IFS (Integrated File System) and is available for use by any authorized PC user with a suitable Client Access network drive assigned.

You can choose several different formats for your files, depending on your particular requirements. Additional formatting options will be introduced in future versions of the product. For the moment the formats supported are:

#### ***Adobe® PDF (Portable Document Format)***

Adobe® Portable Document Format (PDF) is the de facto standard for electronic document distribution. Adobe PDF is a universal file format that preserves the

fonts and formatting of the source document. PDF files are compact and can be shared, viewed, navigated, and printed exactly as intended by anyone with a copy of the free Adobe Acrobat Reader, which can be downloaded from many places on the Internet, including the Adobe site at <http://www.adobe.com/products/acrobat/readstep.html>.

Adobe PDF is the ideal format for electronic document distribution because it overcomes the problems commonly encountered in electronic file sharing. If you create a document in PDF format and e-mail it to a customer, so long as the recipient has a copy of the Acrobat reader, you can be confident that they will be able to read and print the document and that it will appear to them just as it did to you when you created it.

PDF files can be published and distributed anywhere. You can attach them to e-mail, make them available on a corporate server, an Intranet or Extranet, post them on Web sites or circulate them on CD-ROM.

If you want to use **CoolSpools** to distribute and archive your AS/400 reports, PDF is the format to choose.

### ***RTF (Rich Text Format )***

RTF (Rich Text Format) is a format understood and handled by most if not all modern Word Processing applications, including MS Word, Lotus WordPro, MS WordPad etc.

**CoolSpools** preserves text formatting and fonts found in your spooled files and reproduces these in the RTF document.

### ***HTML (Hypertext Mark-up Language)***

However, if for some reason you want to view your reports in a browser, such as Netscape® Navigator or Microsoft® Internet Explorer, **CVTSPLSTMF** is there to help you.

HTML is the language in which web pages are written. **CVTSPLSTMF** can convert your spooled files to HTML and in fact supports not one but two HTML styles. Which of these styles you choose depends on whether you want your text to appear laid out on the screen exactly as it was on the original paper report, or whether you are more interested in ensuring that text attributes such as bold type, fonts and underlining are reflected in the HTML version of the report.

Please note that HTML is not recommended for distributing and archiving your AS/400 reports. HTML has certain limitations which makes it impossible to guarantee that a report converted to HTML will appear on screen or print exactly like the original report. For example, there is no means of controlling page throws in HTML. In the main, you should choose PDF for distributing and archiving your reports.

### ***Excel Format***

**IF you specify TOFMT(\*XLS), CVTSPLSTMF** will create an Excel © spreadsheet from your spooled file. This is most suitable for spooled files

containing columns of text and/or numbers. Options allow you to exclude unwanted lines from your report as it is converted to Excel format (for example, report headings not required in the spreadsheet).

### ***Delimited ASCII Text***

***CVTSPLSTMF*** can also convert your report into a delimited ASCII text file, such as a CSV (comma-separated variable file). This format is ideal for loading reports containing columns of numbers into a spreadsheet, Business Intelligence tool or other application for further manipulation. ***CVTSPLSTMF*** can use any field delimiter you like (by default a comma, but also tabs, blanks etc.) and any string separator you specify (by default a double quote “), allowing you to generate files in the precise format required by your PC application.

### ***Plain ASCII Text***

***CVTSPLSTMF*** will also simply convert your AS/400 spooled file to a basic ASCII text file. This file can then be loaded into virtually any PC application, such as a word processor or spreadsheet.

Please note, however, that when loaded into a spreadsheet, you may need to do additional work to line the text up correctly in columns. You will probably achieve better results using a delimited ASCII text file.

### ***Archive format***

When you specify the TOFMT(\*SAV) option, ***CVTSPLSTMF*** generates a highly compressed archive file from your spooled file.

This facility allows you to save AS/400 resources by removing spooled files from your system. You simply convert the spooled file to a stream file in \*SAV format, then delete the original spooled file. If you need to re-print the report, you can restore it from the archive file using the sister command to CVTSPLSTMF, namely CVTSTMSPLF (Convert Spooled File to Stream File).

Stream files in \*SAV format can also be attached to e-mails and sent to other AS/400 users, who can then restore the spooled file to their own system.

Please note that the stream files created using this option still reside on your AS/400 disks, albeit in the IFS rather than as a spooled file. You should use the OS/400 **SAV** command to back these stream files up to tape, or copy them to CD-ROM or to a PC server, before considering that your spooled files are truly secure. Please note also that these archive files are not viewable in any PC application.

## **Features**

***CVTSPLSTMF*** provides a number of useful features available from few if any other similar products on the market.

## ***Support for AFP, IPDS and SCS spooled files***

Many businesses have moved on from traditional AS/400 line-printer reports and are using AFP (Advanced Function Printing) to improve the presentation of their reports. AFP in particular allows you to incorporate proportional fonts, color and graphics into your reports to improve their appearance, and to use page segments and overlays to simplify programming.

Most other products which convert spooled files are limited to old-fashioned SCS (SNA Character String) spooled files. ***CVTSPLSTMF*** supports AFP and IPDS spooled files as well as SCS. This support comprises: color, line graphics, variable fonts, overlays and page segments, raster graphics and most types of compressed IOCA images.

Please note that there are some less widely used AFP features that are not supported in this release of ***CoolSpools***. These include: bar codes other than Code 3-of-9 and Code 128, IOCA images compressed using compression methods other than the normal IBM MMR algorithm and certain types of page rotation. We hope to introduce full support for these aspects of AFP in a future version of this product.

## ***Colorize your reports!***

If your AFP report includes color, ***CVTSPLSTMF*** will reproduce those colors in your PDF or HTML output.

However, if you would like to add color to your reports but can't afford an AS/400 color printer, ***CVTSPLSTMF*** can still help. You simply specify a color for both the text and background of your report and ***CVTSPLSTMF*** will use those colors in PDF and HTML. Alternatively, you might want to re-code your report to use colors, but print the PDF output on an affordable PC color printer rather than a mega-bucks AS/400 printer.

## ***User-definable Font Mapping***

Where the AS/400 font used in your report is not available to PDF, ***CVTSPLSTMF*** will attempt to choose a suitable alternative font. However, this may not always give acceptable results. No matter: ***CVTSPLSTMF*** provides you with the information you need to specifying how fonts in the report should be mapped, thus ensuring that the resultant PDF document is as close as possible in appearance to the original report.

## ***PDF Bookmarks***

This popular feature of ***CVTSPLSTMF*** allows you to go further than just creating a copy of your report in PDF: you can actually enhance the content of your report too by creating a set of bookmarks. These help your users navigate around the report on screen and go quickly and easily to the information they require.

Bookmarks can be created based on either a piece of text which appears regularly at the same position on each page, or by means of a keyword or phrase

(e.g. "Customer name:") which labels the information you wish to use to create the bookmarks.

### ***PDF Data Compression***

The data in your PDF files is held in compressed format. This can dramatically reduce the size of the files created, making them quicker and easier to e-mail and load, and making PDF even more attractive as an option for archiving your reports.

### ***Web-optimized PDF***

PDF files are "linearized", that is they are structured in a way that minimizes the time it takes to open the file over the Web or your Wide Area Network.

### ***Encrypted and password-protected PDF***

PDF files can now be password protected. A password-protected PDF file will require you to enter a password before it can be opened in Adobe Acrobat Reader, Adobe Acrobat or any other PDF Viewer application.

Moreover, just opening the file in a text editor is to no avail as the textual content of the file is encrypted using a highly secure industry standard encryption algorithm.

### ***Spooled File Splitting***

***CVTSPLSTMF*** has the ability to split a single spooled file into several separate PDF files. Perhaps you have an application which creates a single spooled file containing invoices for a number of customers. You can ask for the spooled file to be split into separate pdf files for each customer. You can ask for splitting to occur every so many pages, or every time a specified key word or phrase appears in the spooled file.

### ***User-definable Exit Program***

You can instruct ***CVTSPLSTMF*** to call a user-written exit program every time it finishes creating a PDF file. The exit program is passed a standard set of parameters, plus a string of text extracted from the spooled file according to criteria that you define. Coupled with the ability to split spooled files described above, this gives you the power to adapt existing applications to the challenge of distributing reports and documents via e-mail rather than snail mail.

For example, you could tell ***CVTSPLSTMF*** to split your invoices spooled file into separate PDF files for each customer, and ask it to call an exit program after each PDF file has been created. ***CVTSPLSTMF*** could extract the customer number from the spooled file and pass it to the exit program as a parameter. The exit program could then retrieve the customer's e-mail address from the database and invoke a mail application to send the pdf file to the customer.

## **Installation**

If you downloaded **CoolSpools** from the Internet, or received the product via e-mail, you will have received a zip file called **cvtspl.zip**. This zip files contain everything you need to get started with **CoolSpools**. This section describes two possible procedures for installing **CoolSpools** on your AS/400. We are hoping to simplify and automate this area in the near future. However, for the moment, please select one of the two methods described below and follow the steps carefully and in the order indicated. Steps to be carried out on the PC are shown in **green**, steps to be carried out on the AS/400 in **blue**.

### **Method 1: Installing CoolSpools using FTP**

This method uses FTP (File Transfer Protocol) to transfer the CoolSpools product files from your PC to your AS/400. It is assumed that you have stored the cvtspl.zip file which you downloaded from the Internet or received via e-mail on your PC and that you have enabled an FTP link between your PC and your AS/400.

#### **Step 1**

**On your PC**, create an install directory for CoolSpools, either within Windows Explorer, or from the DOS prompt, e.g.:

```
md c:\cvtspl
```

Save or copy the zip file that you downloaded or received into this directory.

#### **Step 2**

**On your PC**, unzip the contents of the zip file into the directory you just created using pkunzip, WinZip, Norton UnZip etc.

#### **Step 3**

**On your AS/400**, decide which library you wish to use for the install procedure and create a save file in that library, e.g.

```
crtsavf lib_name/cvtspl
```

where **lib\_name** is the name of the library you chose to use.

#### **Step 4**

**On your PC**, start an FTP session to your AS/400. You may use FTP packages such as CuteFTP or simply go to the DOS prompt and type:

```
ftp AS400_name
```

where AS400\_name is the name of your AS/400.

You will be prompted to enter your user id and password. Please ensure that you sign on with a user id that has adequate authority. We recommend that you sign on as the Security Officer QSECOFR.

Issue the following FTP commands within the FTP session:

### **binary**

This command instructs FTP not to carry out any conversion on the data to be sent to the AS/400.

### **cd lib\_name**

where lib\_name is the name of the library in which you created the save file in Step 3. This command makes lib\_name your current library on the AS/400.

### **lcd dir\_name**

where dir\_name is the name of the directory on your PC in which you unzipped the install files earlier. This command makes dir\_name your current directory on the PC.

### **put cvtspl.sav cvtspl**

This command transmits the contents of the file cvtspl.sav, which was earlier unzipped from cvtspl.zip, and stores them in the save file called cvtspl which you created earlier on your AS/400.

### **Step 5**

**On your AS/400**, ensure that the previous step worked by displaying the contents of save file cvtspl:

### **dspsavf lib\_name/cvtspl**

If you see the message (**No records in save file**) then something has gone wrong during the preceding steps. Try following them again. If you still cannot get this procedure to work, contact us at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

### **Step 6**

**On your AS/400**, issue the following command to restore the CoolSpools licensed program to your system:

```
rstlicpgm      licpgm(2CVTSPL)
                dev(*savf)
                option(*base)
                rstobj(*all)
                lng(*primary)
                output(*none)
                rls(*first)
                replacerls(*only)
                savf(lib_name/cvtspl)
```

where lib\_name is the name of the library in which you created the save file in Step 3.

### **Step 7**

If you have downloaded CoolSpools from the Internet, you can use the product without a license key for up to 30 days. At the end of that period CoolSpools will notify you that it can no longer be used without a license key.

To obtain a license key contact @riadne at [sales@ariadnesoftware.co.uk](mailto:sales@ariadnesoftware.co.uk).

If you have been issued with a license key, proceed as follows.

**On your AS/400**, issue the following command to add a license key to enable you to use the CoolSpools licensed program:

```
addlickey      lickeyinp(*prompt)
                prdid(2CVTSPL)
                licterm(V2R1)
                feature(5001)
                serial(*local)
                prcgrp(Ppp)
                lickey(XXXXXX YYYYYY ZZZZZZ)
                usglmt(*nomax)
                expdate(exp_date)
                vnddta(*none)
```

where: **XXXXXX YYYYYY ZZZZZZ** is the license key you received by e-mail from us

**exp\_date** is the expiry date mentioned in the e-mail

and: **Ppp** is the processor group mentioned in the e-mail

**CoolSpools** is now ready for you to use!

If you have any trouble at any point in this procedure, please contact us by e-mail at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

## **Method 2: Installing CoolSpools using the IFS**

This method uses a Client Access network drive assigned to the IFS of your AS/400 to transfer the CoolSpools product files from your PC to your AS/400. It is assumed that you have stored the cvtspl.zip file (which you downloaded from the Internet or received via e-mail) on your PC.

### **Step 1**

**On your PC**, create an install directory for CoolSpools, either within Windows Explorer, or from the DOS prompt, e.g.:

```
md c:\PC_dir
```

where PC\_dir is the name of the directory you wish to use for this installation.

Save or move the zip file that you downloaded or received into this directory.

### **Step 2**

**On your PC**, unzip the contents of the zip file into the directory you just created using pkunzip, WinZip, Norton UnZip etc.

### **Step 3**

**On your PC**, ensure that you have a network drive assigned to a directory in the root file system of your AS/400's IFS. This can be done in Windows Explorer by selecting Tools, Map Network Drive.

Create a directory in the IFS to receive the CoolSpools files. This can be done in Windows Explorer by selecting File, New, Folder, or at the DOS prompt as follows:

```
md i:\IFS_dir
```

where **I:** is the network drive assigned to your AS/400 and **IFS\_dir** is the name of the directory you wish to use for this installation.

### **Step 4**

Copy the contents of your PC's install directory created in Step 1 above to the IFS directory created in Step 3 above using Windows Explorer, or from the DOS prompt as follows:

```
copy c:\PC_dir\*. * I:\IFS_dir
```

where **PC\_dir** is the name of the directory on your PC, **I:** is the network drive assigned to your AS/400 and **IFS\_dir** is the name of the directory you wish to use for this installation.

### **Step 5**

**On your AS/400**, decide which library you wish to use for the install procedure and create a save file in that library, e.g.

```
crtsave lib_name/cvtspl
```

where **lib\_name** is the name of the library you chose to use.

### **Step 6**

**On your AS/400**, copy the contents of the cvtspl.sav file from the IFS directory to the save file just created, by issuing the following command:

```
copyfromstmf fromstmf('/IFS_dir/cvtspl.sav')  
tombr('/qsys.lib/lib_name.lib/cvtspl.file')  
mbropt(*replace)  
cvtdta(*none)  
endlinfmt(*fixed)  
tabexpn(*no)
```

where **IFS\_dir** is the name of the IFS directory into which you copied the install files in Step 4 above, and **lib\_name** is the name of the library in which you created the save file in Step 5 above.

### **Step 7**

**On your AS/400**, ensure that the previous step worked by displaying the contents of save file cvtspl:

```
dspsavf lib_name/cvtspl
```

where lib\_name is the name of the library in which your created the save file in Step 5.

If you see the message (**No records in save file**) then something has gone wrong during the preceding steps. Try following them again. If you still cannot get this procedure to work, contact us at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

### **Step 8**

**On your AS/400**, issue the following command to restore the CoolSpools licensed program to your system:

```
rstlicpgm      licpgm(2CVTSPL)  
               dev(*savf)  
               option(*base)  
               rstobj(*all)  
               lng(*primary)  
               output(*none)  
               rls(*first)  
               replacerls(*only)  
               savf(lib_name/cvtspl)
```

where lib\_name is the name of the library in which your created the save file in Step 5.

### **Step 9**

If you have downloaded CoolSpools from the Internet, you can use the product without a license key for up to 30 days. At the end of that period CoolSpools will notify you that it can no longer be used without a license key.

To obtain a license key contact @riadne at [sales@ariadnesoftware.co.uk](mailto:sales@ariadnesoftware.co.uk). If you have been issued with a license key, proceed as follows.

**On your AS/400**, issue the following command to add a license key to enable you to use the CoolSpools licensed program:

```
addlickey      lickeyinp(*prompt)  
               prdid(2CVTSPL)  
               licterm(V2R1)
```

**feature(5001)  
serial(\*local)  
prcgrp(Ppp)  
lickey(XXXXXX YYYYYY ZZZZZZ)  
usglmt(\*nomax)  
expdate(exp\_date)  
vnddta(\*none)**

where: **XXXXXX YYYYYY ZZZZZZ** is the license key you received by e-mail from us

**exp\_date** is the expiry date mentioned in the e-mail

and: **Ppp** is the processor group mentioned in the e-mail

**CoolSpools** is now ready for you to use!

If you have any trouble at any point in this procedure, please contact us by e-mail at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

## **Maintenance**

Periodically we will release Program Temporary Fixes (PTFs) to resolve any problems found with CoolSpools.

PTFs are supplied in the form of an AS/400 save file stored within a zip file, and can be downloaded from the Bug Fixes section of the @riadne web site at <http://www.ariadnesoftware.co.uk>.

The procedure for loading and applying a PTF for CoolSpools is as follows.

First download the PTF you require from the Internet. The PTF will be held in a file called Q2CVnnnn.zip, where nnnn is the PTF id.

This section describes two possible procedures for copying the AS/400 save file held in this zip file to your AS/400. We are hoping to simplify and automate this area in the near future. However, for the moment, please select one of the two methods described below and follow the steps carefully and in the order indicated. Steps to be carried out on the PC are shown in **green**, steps to be carried out on the AS/400 in **blue**.

### **Method 1: Installing a PTF using FTP**

This method uses FTP (File Transfer Protocol) to transfer the PTF file from your PC to your AS/400. It is assumed that you have stored the zip file which you downloaded from the Internet on your PC and that you have enabled an FTP link between your PC and your AS/400.

#### **Step 1**

**On your PC**, create a directory for the PTF installation, either within Windows Explorer, or from the DOS prompt, or re-use the directory you used to install originally, e.g.:

```
md c:\cvtspl
```

Save or copy the zip file that you downloaded or received into this directory.

#### **Step 2**

**On your PC**, unzip the contents of the zip file into the directory you just created using pkunzip, WinZip, Norton UnZip etc.

#### **Step 3**

**On your AS/400**, first sign off then sign back on again to ensure that you do to have any CoolSpools objects in your QTEMP library. Then decide which library you wish to use for the PTF install procedure and create a save file in that library. The name of this save file should be Q2CVnnnn, where nnnn is the PTF id, e.g.

```
crtsave lib_name/Q2CV0001
```

where **lib\_name** is the name of the library you chose to use.

#### **Step 4**

**On your PC**, start an FTP session to your AS/400. You may use FTP packages such as CuteFTP or simply go to the DOS prompt and type:

**ftp AS400\_name**

where AS400\_name is the name of your AS/400.

You will be prompted to enter your user id and password. Please ensure that you sign on with a user id that has adequate authority. We recommend that you sign on as the Security Officer QSECOFR.

Issue the following FTP commands within the FTP session:

**binary**

This command instructs FTP not to carry out any conversion on the data to be sent to the AS/400.

**cd lib\_name**

where lib\_name is the name of the library in which you created the save file in Step 3. This command makes lib\_name your current library on the AS/400.

**lcd dir\_name**

where dir\_name is the name of the directory on your PC in which you unzipped the install files earlier. This command makes dir\_name your current directory on the PC.

**put Q2CVnnnn.sav Q2CVnnnn**

This command transmits the contents of the file Q2CV0001.sav, which was earlier unzipped from Q2CV0001.zip, and stores them in the save file called Q2CVnnnn which you created earlier on your AS/400.

#### **Step 5**

**On your AS/400**, ensure that the previous step worked by displaying the contents of save file Q2CVnnnn:

**dspsavf lib\_name/Q2CVnnnn**

If you see the message **(No records in save file)** then something has gone wrong during the preceding steps. Try following them again. If you still cannot get this procedure to work, contact us at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

#### **Step 6**

**On your AS/400**, issue the following command to load the PTF on to your system:

**lodptf                      licpgm(2CVTSPL)  
                                  dev(\*savf)**

```
select(2CVnnnn)
savf(lib_name/Q2CVnnnn)
```

where lib\_name is the name of the library in which you created the save file in Step 3 and nnnn is the PTF id.

### **Step 7**

On your AS/400, issue the following command to apply the PTF to your system:

```
apypf          licpgm(2CVTSPL)
               select(2CVnnnn)
               apy(*perm)
```

where: nnnn is the PTF

If you have any trouble at any point in this procedure, please contact us by e-mail at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

## **Method 2: Installing a PTF using the IFS**

This method uses a Client Access network drive assigned to the IFS of your AS/400 to transfer the PTF from your PC to your AS/400. It is assumed that you have stored the PTF zip file (which you downloaded from the Internet or received via e-mail) on your PC.

### **Step 1**

On your PC, create an install directory for the PTF, either within Windows Explorer, or from the DOS prompt, or use the directory you created for the original installation, e.g.:

```
md c:\PC_dir
```

where PC\_dir is the name of the directory you wish to use for this installation.

Save or move the zip file that you downloaded or received into this directory.

### **Step 2**

On your PC, unzip the contents of the zip file into the directory you just created using pkunzip, WinZip, Norton UnZip etc.

### **Step 3**

On your PC, ensure that you have a network drive assigned to a directory in the root file system of your AS/400's IFS. This can be done in Windows Explorer by selecting Tools, Map Network Drive.

Create a directory in the IFS to receive the PTF. This can be done in Windows Explorer by selecting File, New, Folder, or at the DOS prompt as follows:

```
md i:\IFS_dir
```

where **I:** is the network drive assigned to your AS/400 and **IFS\_dir** is the name of the directory you wish to use for this installation.

#### **Step 4**

Copy the contents of your PC's PTF directory created in Step 1 above to the IFS directory created in Step 3 above using Windows Explorer, or from the DOS prompt as follows:

```
copy c:\PC_dir\*. * I:\IFS_dir
```

where **PC\_dir** is the name of the directory on your PC, **I:** is the network drive assigned to your AS/400 and **IFS\_dir** is the name of the directory you wish to use for this installation.

#### **Step 5**

**On your AS/400**, first sign off then sign back on again to ensure that you do to have any CoolSpools objects in your QTEMP library. Then decide which library you wish to use for the install procedure and create a save file in that library. The save file should be called **Q2CVnnnn**, where **nnnn** is the PTF id, e.g.

```
crtsavf lib_name/Q2CVnnnn
```

where **lib\_name** is the name of the library you chose to use

and **nnnn** is the PTF id.

#### **Step 6**

**On your AS/400**, copy the contents of the **Q2CVnnnn.sav** file (where **nnnn** is the PTF id) from the IFS directory to the save file just created, by issuing the following command:

```
copyfrmstmf      fromstmf('/IFS_dir/Q2CVnnnn.sav')  
tombr('/qsys.lib/lib_name.lib/Q2CVnnnn.file')  
mbropt(*replace)  
cvtdta(*none)  
endlinfmt(*fixed)  
tabexpn(*no)
```

where **IFS\_dir** is the name of the IFS directory into which you copied the PTF files in Step 4 above, and **lib\_name** is the name of the library in which you created the save file in Step 5 above.

#### **Step 7**

**On your AS/400**, ensure that the previous step worked by displaying the contents of save file **Q2CVnnnn**:

```
dspsavf lib_name/Q2CVnnnn
```

If you see the message (**No records in save file**) then something has gone wrong during the preceding steps. Try following them again. If you still cannot get this procedure to work, contact us at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

### **Step 8**

**On your AS/400**, issue the following command to load the PTF on to your system:

```
lodptf          licpgm(2CVTSPL)
                dev(*savf)
                select(2CVnnnn)
                savf(lib_name/Q2CVnnnn)
```

where lib\_name is the name of the library in which you created the save file in Step 3 and nnnn is the PTF id.

### **Step 9**

**On your AS/400**, issue the following command to apply the PTF to your system:

```
apyptf          licpgm(2CVTSPL)
                select(2CVnnnn)
                apy(*perm)
```

where: **nnnn** is the PTF

If you have any trouble at any point in this procedure, please contact us by e-mail at [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk).

## **Using CVTSPLSTMF**

### **Getting Started**

In order to convert your spooled files, you need to run the CVTSPLSTMF command. After you have installed **CoolSpools**, this command will exist in a library also called **CVTSPLSTMF**. Before running the CVTSPLSTMF command ensure that this library is in your library list by issuing the following command:

**ADDLIBLE CVTSPLSTMF**

#### **N.B.**

On some systems, the CVTSPLSTMF library may restore with PUBLIC(\*EXCLUDE) authority. You may need to grant, or ask your system administrator to grant, PUBLIC(\*USE) authority to the CVTSPLSTMF library as follows:

**GRTOBJAUT CVTSPLSTMF \*LIB PUBLIC(\*USE)**

before you can start to use CoolSpools.

Also, if your spooled file contains references to any overlay, font resource or page segment objects, these must also be located through your library list. You should also add the library or libraries in which these objects are located to your library list before running **CVTSPLSTMF**, like this:

**ADDLIBLE library\_name**

where **library\_name** is the name of the library in which the object is located.

### **Command Parameters**

The sections below look at each of the parameters to the **CVTSPLSTMF** command in turn and explain how they should be used to get the most out of this invaluable utility.

In the examples, an ellipsis (...) indicates that a number of required parameters have been omitted for the sake of clarity.

#### **FROMFILE**

The **FROMFILE** (From File) parameter specifies the name of the spooled file that is to be converted to a stream file.

**Example: CVTSPLSTMF FROMFILE(QSYSPRT)...**

Here the CVTSPLSTMF command is being applied to a spooled file called **QSYSPRT**.





**INVOICES**, run by the System Operator **QSYSOPR**, with job number **123456**. The spooled file to be converted was the third spooled file open by the job.

## **TOFMT**

The **TOFMT** (To Format) parameter allows you to select the format to which the spooled file should be converted.

The benefits and relative merits of the different formats are discussed in the Document Formats section of the Introduction to this User Guide.

The options are

- |              |   |
|--------------|---|
| <b>*TEXT</b> | <b>Plain ASCII text.</b><br><br>The textual content of the spooled file is converted to ASCII text records terminated by carriage return/line feed pairs.   |
| <b>*HTXT</b> | <b>HTML Text</b><br><br>Textual data in the spooled file is converted to HTML format, viewable in a browser. However, no attempt is made to represent variable fonts, highlighting or other text attributes.<br><br>Use this format if you wish to view traditional, simple AS/400 reports in a browser, and column alignment and text positioning are more important to you than the precise appearance of the report text.  |
| <b>*HTML</b> | <b>HTML</b><br><br>The contents and attributes of the spooled file are converted to HTML format, viewable in a browser. Text attributes such as variable font sizes and typefaces, highlighting and underlining are reflected in the stream file, but text positioning may vary from the original report.<br><br>Use this format if you wish to view more sophisticated AS/400 reports in a browser, and column alignment and text positioning are less important to you than the style and attributes of the report text |
| <b>*PDF</b>  | <b>Portable Document Format (PDF)</b><br><br>The contents and attributes of the spooled file are converted to PDF format. PDF files can be viewed and printed using the free Adobe Acrobat Reader program.  |

Every effort is made to ensure that the report will appear in the Acrobat Reader exactly the same as the original paper report.

**\*RTF**

**Rich Text Format (RTF)**

The contents and attributes of the spooled file are converted to RTF format. RTF files can be viewed and printed using most commonly used Word Processing applications, such as MS Word, Lotus WordPro or MS WordPad.

**\*XLS**

**Excel © (BIFF) format**

The textual content of the spooled file is converted to columns and rows in an Excel spreadsheet.

**\*CSV**

**Delimited ASCII text**

The textual content of the spooled file is converted to delimited ASCII text records terminated by carriage return/line feed pairs. The delimiters used to separate fields and enclose strings are defined on the **DELIMITERS** parameters. This option enables you to create files in CSV (comma-separated variable) format and other similar formats for loading into spreadsheets and other PC applications.

**\*SAV**

**Archive**

The spooled file is saved into a compressed archive format. The spooled file can later be restored using the CVTSTMSPLF (Convert Stream File to Spooled File) command.

**Example:** **CVTSPLSTMF FROMFILE(INVOICES)...  
TOFMT(\*PDF)**

Here the CVTSPLSTMF command is being applied to a spooled file called **INVOICES** in order to create a stream file in PDF format.

**STMFOPT**

The **STMFOPT** (Stream File Option) parameter allows you to select the action CVTSPLSTMF should take if the stream file you have specified on the **TOSTMF** and **TODIR** parameters already exists. This parameter is ignored if the stream file does not already exist.

The options are:

- \*NONE** The command reports an error and the existing file is not changed. For safety's sake, this is the default value.
- \*REPLACE** The existing file is replaced.
- \*ADD** The contents of the spooled file are appended to the existing file. Please note that since PDF files cannot be appended to in this way without becoming unusable, this option is not supported in conjunction with the TOFMT(\*PDF) option.

**Example:** *CVTSPLSTMF FROMFILE(INVOICES)...  
TOFMT(\*TEXT)  
STMFOPT(\*ADD)*

Here the CVTSPLSTMF command is being applied to a spooled file called **INVOICES**. The spooled file is being converted to ASCII text form and the contents of the spooled file will be appended to the end of the existing file.

### **TITLE**

The **TITLE** parameter allows you to define a title for the report. This parameter is available only if something other than **TOFMT(\*TEXT)** is selected.

If you specify TOFMT(\*HTML) or TOFMT(\*HTXT), the text you key in the TITLE parameter will appear in your browser's title bar.

If you specify TOFMT(\*PDF), the text in the TITLE parameter appears in Adobe Acrobat Reader by selecting: File, Document Info & General from the Reader menu.

Alternatively, you may select one of the special values:

- \*NONE** (The default). The report has no title.
- \*STMFILE** The report title is the same as the stream file name specified on the TOSTMF parameter.

**Example:** *CVTSPLSTMF FROMFILE(SALES)...  
TOFMT(\*PDF)  
TITLE('Sales Statistics April 2001')*

The sales report is converted to PDF format. Users can check the Document Information panel in Adobe Acrobat to see the title 'Sales Statistics April 2001' to confirm the nature of the report.

### **PAGESIZE**

The **PAGESIZE** parameter allows you to specify the page size that should be used when displaying or printing a PDF file using Adobe Acrobat. This parameter is available only if **TOFMT(\*PDF)** or **TOFMT(\*RTF)** is selected.

This parameter consists of four parts, the page size, the orientation, and whether or not to implement text and page rotation.

However, there are two special options which are entered as a single value on this command rather than as a pair of values.

Options for the page size are:

- \*SPLF** (The default). CVTSPLSTMF will reproduce the page size of the original spooled file. In the case of an AFP or IPDS spooled file containing several different page sizes or orientations, these will all be reflected on the corresponding pages of the PDF or RTF document.
- \*CUSTOM** You will specify the precise page size on the **CUSTOMPAGE** parameter discussed later. This option is useful if you want to use a paper size not provided as one of the standard options listed below.

The other options for the page size are a variety of commonly used paper sizes. These must be selected in conjunction with a value for the orientation.

- \*A4** 297 x 210 mm.
- \*A5** 210 x 148 mm.
- \*LEGAL** 14 x 8.5 in.
- \*LETTER** 11 x 8.5 in.
- \*EXEC** 10.5 x 7.25 in.

These dimensions assume a landscape orientation. However, the orientation can be altered using the second part of this parameter, the options for which are:

- \*LANDSCAPE** Landscape mode
- \*PORTRAIT** Portrait mode

The next two elements control whether CVTSPLSTMF should implement text and/or page rotation in the spooled file. In some spooled files, a combination of text and page rotation may result in undesirable results which can be improved by suppressing text and/or page rotation.

- \*YES** Implement rotation
- \*NO** Do not implement rotation

Note that if your spooled file has the attribute PAGRTT(\*AUTO), automatic page rotation will occur when the spooled file is printed on a printer and the spooled file does not fit on the page in its standard orientation. For example, if you have a spooled file which is in landscape mode, and you print it to a printer which prints in A4 portrait mode normally, PAGRTT(\*AUTO) will cause the spooled file to be rotated automatically through 90 degrees and printed in landscape mode on the A4 portrait printer.

CoolSpools does not implement an automatic page rotation when PAGRTT(\*AUTO) is in force, as this may or may not be appropriate depending

on the printer for which the spooled file was intended, which CoolSpools cannot know.

However, if you are relying on a page rotation, it is possible that page segments and images contained in the spooled file will appear misplaced and rotated in the PDF files you create using the default parameters. In order to replicate the effects of an automatic page rotation in your spooled file, you need to specify \*YES on the "Implement page rotation" element of this parameter. This will ensure that images and page segments are handled as if the page had been rotated.

**Example:** *CVTSPLSTMF FROMFILE(SALES)...  
TOFMT(\*PDF)  
PAGESIZE(\*A4 \*PORTRAIT)*

The sales report is converted to PDF format. It will appear in Acrobat in A4 portrait format.

### **CUSTOMPAGE**

The **CUSTOMPAGE (Custom Page Size)** parameter only appears if **TOFMT(\*PDF)** or **TOFMT(\*RTF)** is selected and you have specified **\*CUSTOM** on the **PAGESIZE** parameter. This parameter allows you to define a non-standard page size to be used by PDF or RTF.

There are three elements to this parameter: the width of the page, the length of the page, and the unit of measurement.

Options for the unit of measurement are:

<b>*MM</b>	Millimeters
<b>*INCH</b>	Inches

Please note that the program makes no attempt to shrink or expand the report to fit the specified page size.

**Example:** *CVTSPLSTMF FROMFILE(SALES)...  
PAGESIZE(\*CUSTOM)  
CUSTOMPAGE(420 297 \*MM)*

The spooled file is converted to PDF format using a page size of 420 x 297 millimeters (A3).

### **MARGINS**

The **MARGINS (Additional margins)** parameter only appears if **TOFMT(\*PDF)** is selected. This parameter allows you to specify an additional left and/or top margin to be applied to the report. This can be useful where the report when printed appears too close to the left or top edge of the paper. AS/400 reports are frequently designed to fit snug to the left or top edge of the page, whereas many PC printers are unable to print close to the edges of the page.



## **PASSWORD**

The PASSWORD (PDF passwords) parameter only appears if TOFMT(\*PDF) is selected. This parameter allows you to password-protect your PDF files. Password protecting a PDF file allows you to e-mail it safe in the knowledge that, if the e-mail goes astray or is intercepted, it will not be possible to open it without the necessary password. Similarly, sensitive business documents can be stored safely on your company server and will not be accessible by anyone who has not been given the passwords to open them.

PDF passwords are implemented using Adobe's standard encryption method. This highly secure encryption technique employs the MD5 message-digest algorithm (described in Internet RFC 1321, The MD5 Message-Digest Algorithm) and the public-domain ArcFour encryption algorithm.

PDF's standard encryption handler limits the encryption key to 5 bytes (40 bits) in length, in accordance with U.S. cryptographic export requirements.

A PDF file may be allocated a "user" password and an "owner" password.

The "owner" password gives full access to all features of the document, i.e. entering the "owner" password in Adobe Acrobat will enable you to modify, copy, print and annotate the document if you have the necessary software (e.g. Adobe Acrobat).

The "user" password gives either full access or limited access to the document, depending on the user privileges that were granted when the file was created.

The privileges that can be granted are:

- whether the document may be printed
- whether text in the document may be copied
- whether the document can be modified
- whether notes can be added to the document

PLEASE NOTE THAT PASSWORDS ARE CASE-SENSITIVE.

If you forget your password, you will not be able to open your document.

The supplier of this software takes no responsibility for documents that cannot be opened as a result of a lost or forgotten password and has no means to recover documents that have become unusable as a result.

There are 7 elements to this parameter.

The first element indicates whether you wish to password protect the document.

Options are:

- |             |   |
|-------------|---|
| <b>*NO</b>  | The PDF file will not be password protected (default) |
| <b>*YES</b> | At least a user password is needed to open the file   |

The second element is the user password. If **\*YES** is specified for the previous element, a user password must be entered (cannot be left blank).

The password can be any string of characters and numbers. The minimum length is 1 character and the maximum is 32.

The third element is the owner password. If no owner password is entered (i.e. it is left blank), the document will not have an owner password. This means that it will not be possible for anyone to perform any actions not permitted according to the user rights defined in the following parameters.

The owner password can be any string of characters and numbers. The minimum length is 1 character and the maximum is 32.

The remaining four elements to this parameter control the rights granted when the document is opened by entering the "user" password.

These are:

- Allow printing?
- Allow modifications?
- Allow copying of text?
- Allow annotation?

All of these parameters take the following form:

- |             |                                   |
|-------------|-----------------------------------|
| <b>*YES</b> | The action is permitted (default) |
| <b>*NO</b>  | The action is not permitted       |

## **FONT**

The **FONT** parameter only appears if **TOFMT(\*PDF)**, **TOFMT(\*RTF)** or **TOFMT(\*HTML)** is selected. This parameter allows you to specify the way in which fonts used in the spooled file are converted to PC fonts in the HTML, RTF or PDF files created by CVTSPLSTMF.

The default value is **\*CONVERT**. This option instructs CVTSPLSTMF to attempt to map the AS/400 fonts used in the spooled file to a common equivalent PC font. It does this by using an internal table of AS/400 font ids and their attributes,

and by relying on font resources being named according to IBM's naming convention for fonts, character sets and code pages.

The way in which fonts are implemented on the AS/400 is rather different than the way fonts work in most Windows-based PC applications, and any mapping from one to the other is at best approximate. If you select **FONT(\*CONVERT)**, CVTSPLSTMF will use as much information as is available to it to select a suitable font, but this does not always give satisfactory results.

However, CVTSPLSTMF will also assist you in improving the appearance of your reports in PDF, RTF and HTML format by providing messages in the job log to inform you about the font mappings it has made. If you are not satisfied with the appearance of your report in PDF, RTF or HTML, you should examine the job log of the job in which the CVTSPLSTMF command was executed, and locate any messages of the form:

Courier 10 substituted for font id 11

or Courier Bold 10 substituted for font resource C0S0CB10

These messages are intended to help you identify which font is and font resource names need to be mapped. You can try alternative font mappings through the user-definable font mapping facility, implemented via the **CVTFONTID** and **CVTFNTRSC** parameters described below.

All other possible selections for this parameter consist of a font face name and a font size in points.

The possible font typefaces supported are those defined as standard fonts by the PDF specification, namely:

<b>*COURIER</b>	Courier
<b>*COURIERB</b>	Courier Bold
<b>*COURIERO</b>	Courier Oblique
<b>*COURIERBO</b>	Courier Bold Oblique
<b>*HELVETICA</b>	Helvetica
<b>*HELVB</b>	Helvetica Bold
<b>*HELVO</b>	Helvetica Oblique
<b>*HELVB</b>	Helvetica Bold
<b>*HELVBO</b>	Helvetica Bold Oblique
<b>*TIMES</b>	Times Roman
<b>*TIMESB</b>	Times Roman Bold
<b>*TIMESI</b>	Times Roman Italic
<b>*TIMESBI</b>	Times Roman Bold Italic
<b>*SYMBOL</b>	Symbol
<b>*DINGBATS</b>	Zapf Dingbats

These fonts are guaranteed by Acrobat to be available for use when you view a PDF file.

When you select one of these standard fonts, you must also indicate the font size you want (measured in points).

The default font size is **\*CALC**. This special value tells CVTSPLSTMF to try to determine the font size based on the spooled file's width and CPI (characters per inch) value. However, if this does not give the desired results, you can specify a size yourself.

**Example:** **CVTSPLSTMF FROMFILE(SALES)...**  
**TOFMT(\*PDF)**  
**FONT(\*CONVERT)**

Here the sales report is converted to PDF format and CVTSPLSTMF will attempt to select suitable replacement fonts for those used in the spooled file.

**Example:** **CVTSPLSTMF FROMFILE(SALES)...**  
**TOFMT(\*PDF)**  
**FONT(\*HELVETICA 12)**

Here the same report is converted, but Helvetica 12-point font will be used throughout.

## **INCLFILE**

The **INCLFILE** parameter only appears if **TOFMT(\*PDF)** is selected. This parameter allows you to specify up to 20 stream files, located on the AS/400 on which the command is being run, which are to be combined with the spooled files data when generating a PDF file. This option can be used, for example, to incorporate a scanned image of your pre-printed stationery into the PDF file so that it reproduces the appearance of the printed form exactly. Alternatively, you might wish to include your company logo in a PDF file created from an \*SCS spooled file, which cannot use graphics.

At present the included image file(s) **MUST** be an image in JPEG format (.JPG, .JPEG). CVTSPLSTMF must be able to locate the image file in the IFS at the time when the command is run.

The default value is the single value **\*NONE**, which indicates that no such files are to be included.

There are nine elements that must be specified for each included file:

- The IFS path name identifying the file to be included. CVTSPLSTMF will use this path to locate the file at the time the command is executed. If it cannot be located, or if the file is not in JPEG format, an error will be reported.
- The method of inclusion. There are two possibilities: **\*EMBEDDED** and **\*EXTERNAL**. The **\*EMBEDDED** option actually embeds or combines the JPEG file in the PDF file being created. This has the advantage of leaving you with just a single file to manage or distribute, but will increase the size of the resulting PDF file. With the other option (**\*EXTERNAL**), the PDF just contains a link or reference to the JPEG file, which is then brought in by the PDF viewer application when the PDF file is opened. This minimizes the



**Example:** **CVTSPLSTMF FROMFILE(SALES)...**  
**TOFMT(\*PDF)**  
**INCLFILE( ('/images/salesform.jpg'**  
**\*EXTERNAL**  
**\*ALL**  
**0 0 \*INCH**  
**form.jpg \*DOS)**

Here the command achieves exactly the same results, except that the JPG is not embedded in the PDF file. Instead, when the PDF file is opened, the viewer application (which is intended to run on a DOS/Windows platform) will look for a file called **form.jpg**. Since just a simple file name is specified, the viewer application will look for the **form.jpg** file in the same directory where the PDF file is located.

## **COLOR**

The **COLOR** parameter only appears if **TOFMT(\*PDF)** or **TOFMT(\*HTML)** is selected.

There are two elements to this parameter.

The first element (Text color) allows you to specify the color to be used for black text in the original spooled file. Any text in the spooled file which is not black will retain its original color. Any black text will assume the color specified here instead.

Options are:

<b>*BLACK</b>	Black
<b>*RED</b>	Red
<b>*BLUE</b>	Blue
<b>*GREEN</b>	Green
<b>*YELLOW</b>	Yellow
<b>*WHITE</b>	White
<b>*MAGENTA</b>	Magenta
<b>*CYAN</b>	Cyan
<b>*BROWN</b>	Brown
<b>*GRAY</b>	Gray
<b>*GRAY</b>	Gray
<b>*PINK</b>	Pink
<b>*PALEBLUE</b>	Pale blue
<b>*PALEGREEN</b>	Pale green
<b>*PALEYLW</b>	Pale yellow
<b>*PALEMAG</b>	Pale magenta
<b>*PALECYAN</b>	Pale cyan
<b>*PALEBROWN</b>	Pale brown
<b>*PALEGRAY</b>	Pale gray

The second element (Background color) allows you to specify the color to be used for the report background.

Options are:

<b>*BLACK</b>	Black
<b>*RED</b>	Red
<b>*BLUE</b>	Blue
<b>*GREEN</b>	Green
<b>*YELLOW</b>	Yellow
<b>*WHITE</b>	White
<b>*MAGENTA</b>	Magenta
<b>*CYAN</b>	Cyan
<b>*BROWN</b>	Brown
<b>*GRAY</b>	Gray
<b>*GRAY</b>	Gray
<b>*PINK</b>	Pink
<b>*PALEBLUE</b>	Pale blue
<b>*PALEGREEN</b>	Pale green
<b>*PALEYLW</b>	Pale yellow
<b>*PALEMAG</b>	Pale magenta
<b>*PALECYAN</b>	Pale cyan
<b>*PALEBROWN</b>	Pale brown
<b>*PALEGRAY</b>	Pale gray

**Example:** **CVTSPLSTMF FROMFILE(SALES)...**  
**TOFMT(\*PDF)**  
**COLOR(\*BLUE \*PALEYLW)**

The sales report is converted to PDF. Any black text in the report will appear blue in Adobe Acrobat. Other colored text will retain its original color. The text will appear against a pale yellow background.

### ***DFTFONT***

The **DFTFONT** (Default Font) parameter only appears if **TOFMT(\*PDF)**, **TOFMT(\*RTF)** or **TOFMT(\*HTML)** is selected. This parameter allows you to specify a default font to be used where CVTSPLSTMF is unable to identify a suitable font to use.

For example, if your report uses a named AS/400 font resource, and the name of that font resource does not conform to IBM's naming standards, and you have not specified a font conversion on the **CVTFNTRSC** parameter, CVTSPLSTMF will use the default font you specify here as a replacement for the unknown AS/400 font resource.

All other possible selections for this parameter consist of a font face name and a font size in points. The possible font typefaces supported are those defined as standard fonts by the PDF specification, namely:

<b>*COURIER</b>	<b>Courier</b>
<b>*COURIERB</b>	<b>Courier Bold</b>

<b>*COURIERO</b>	<b>Courier Oblique</b>
<b>*COURIERBO</b>	<b>Courier Bold Oblique</b>
<b>*HELVETICA</b>	<b>Helvetica</b>
<b>*HELVB</b>	<b>Helvetica Bold</b>
<b>*HELVO</b>	<b>Helvetica Oblique</b>
<b>*HELVB</b>	<b>Helvetica Bold</b>
<b>*HELVBO</b>	<b>Helvetica Bold Oblique</b>
<b>*TIMES</b>	<b>Times Roman</b>
<b>*TIMESB</b>	<b>Times Roman Bold</b>
<b>*TIMESI</b>	<b>Times Roman Italic</b>
<b>*TIMESBI</b>	<b>Times Roman Bold Italic</b>
<b>*SYMBOL</b>	<b>Symbol</b>
<b>*DINGBATS</b>	<b>Zapf Dingbats</b>

These fonts are guaranteed by Acrobat to be available for use when you view a PDF file.

The default is Courier 10-point.

When you select one of these standard fonts, you must also indicate the font size you want (measured in points).

**Example:** **CVTSPLSTMF FROMFILE(SALES)...**  
**TOFMT(\*PDF)**  
**FONT(\*CONVERT)**  
**DFTFONT(\*HELVETICA 12)**

Here the sales report is converted to PDF format, and CVTSPLSTMF is instructed to map AS/400 fonts to suitable substitute fonts recognized by Acrobat. Where CVTSPLSTMF cannot identify a suitable substitute font, Helvetica 12-point will be used.

You must also specify a font size.

If you have also selected **TOFMT(\*HTML)**, the font size you indicate here has a further significance. HTML font sizes are defined relative to a base font size. The font size defined on the DFTFONT parameter serves as the base font size for HTML (i.e. it is mapped to Fontsize="3" in the HTML) and all other fonts will appear large or small relative to this font.

For example, if you specify a default font size of 12 on this parameter, a font size of 10 in the spooled file will appear as a small font in the resultant html (since 10 < 12).

However, if you specify a default font size of 8 on this parameter, a font size of 10 in the spooled file will appear as a large font in the HTML (since 10 > 8).

**Example: CVTSPLSTMF FROMFILE(SALES)...  
TOFMT(\*HTML)  
FONT(\*CONVERT)  
DFTFONT(\*HELVETICA 12)**

The sales report is converted to HTML format. CVTSPLSTMF attempts to map AS/400 fonts to substitute PC fonts, and where this is not possible, Helvetica is used. 12-point is defined as being the basic HTML font size, and all other fonts in the report will appear large or small relative to this font.

### **CVTFONTID**

The **CVTFONTID** (Convert Font Id) parameter only appears if **TOFMT(\*PDF)**, **TOFMT(\*RTF)** or **TOFMT(\*HTML)** is selected. This parameter allows you to define your own mappings for those fonts used in the spooled file that are identified by a font id number. You may define mappings here in order to improve the appearance of your report in PDF, RTF or HTML format where **FONT(\*CONVERT)** alone does not give satisfactory results. Up to 100 mappings may be defined.

The default is **\*NONE**, which indicates that no user-definable font id mappings are specified.

Other than **\*NONE**, all other selections for this parameter consist of three elements: a font identifier, a typeface name and a font size (in points).

Where the specified font id appears in the spooled file, it will be mapped to the font typeface and point size specified here in the resultant PDF or HTML file.

CVTSPLSTMF notifies you of the font mappings that it has chosen by sending a message to the job log when you run the command. If you are not satisfied with the appearance of your spooled file in the stream file, consider using this parameter or the CVTFNTRSC parameter to define your own font mappings. Refer to the discussion of the **FONT** parameter above for further information about how CVTSPLSTMF assists you in identifying the font ids that need to be mapped.

Enter the id of the font to be mapped, plus the name of the typeface to be substituted, and its point size.

The following typefaces are available for selection:

<b>*COURIER</b>	<b>Courier</b>
<b>*COURIERB</b>	<b>Courier Bold</b>
<b>*COURIERO</b>	<b>Courier Oblique</b>
<b>*COURIERBO</b>	<b>Courier Bold Oblique</b>
<b>*HELVETICA</b>	<b>Helvetica</b>
<b>*HELVB</b>	<b>Helvetica Bold</b>
<b>*HELVO</b>	<b>Helvetica Oblique</b>
<b>*HELVB</b>	<b>Helvetica Bold</b>
<b>*HELVBO</b>	<b>Helvetica Bold Oblique</b>
<b>*TIMES</b>	<b>Times Roman</b>

<b>*TIMESB</b>	<b>Times Roman Bold</b>
<b>*TIMESI</b>	<b>Times Roman Italic</b>
<b>*TIMESBI</b>	<b>Times Roman Bold Italic</b>
<b>*SYMBOL</b>	<b>Symbol</b>
<b>*DINGBATS</b>	<b>Zapf Dingbats</b>

You must also specify a font size in points. The default is 10-point.

**Example:** **CVTSPLSTMF FROMFILE(SALES)  
TOSTMF(sales.pdf)  
TODIR('/sales/2001/april')  
TOFMT(\*PDF)  
FONT(\*CONVERT)  
CVTFONTID(11 \*COURIER 10)**

The sales report is converted to PDF format. CVTSPLSTMF will attempt to map fonts, but font id 11 is explicitly mapped to Courier 10-point.

### **CVTFNTRSC**

The **CVTFNTRSC** (Convert Font Resource) parameter only appears if **TOFMT(\*PDF)**, **TOFMT(\*RTF)** or **TOFMT(\*HTML)** is selected. This parameter allows you to define your own mappings those for fonts used in the spooled file that are identified by a font resource name. You may define mappings here in order to improve the appearance of your report in PDF, RTF or HTML format if **FONT(\*CONVERT)** alone does not give satisfactory results. Up to 100 mappings may be defined.

The default is **\*NONE**, which indicates that no user-definable font resource mappings are specified.

Other than **\*NONE**, all other selections for this parameter consist of three elements: a font resource name, a typeface name and a font size (in points).

Where the specified font resource name appears in the spooled file, it will be mapped to the font typeface and point size specified here in the resultant PDF or HTML file.

CVTSPLSTMF notifies you of the font mappings that it has chosen by sending a message to the job log when you run the command. If you are not satisfied with the appearance of your spooled file in the stream file, consider using this parameter or the CVTFONTID parameter to define your own font mappings. Refer to the discussion of the **FONT** parameter above for further information about how CVTSPLSTMF assists you in identifying the font ids that need to be mapped.

Enter the name of the font resource to be mapped, plus the name of the typeface to be substituted, and its point size.

The following typefaces are available for selection:

<b>*COURIER</b>	<b>Courier</b>
<b>*COURIERB</b>	<b>Courier Bold</b>

<b>*COURIERO</b>	Courier Oblique
<b>*COURIERBO</b>	Courier Bold Oblique
<b>*HELVETICA</b>	Helvetica
<b>*HELVB</b>	Helvetica Bold
<b>*HELVO</b>	Helvetica Oblique
<b>*HELVB</b>	Helvetica Bold
<b>*HELVBO</b>	Helvetica Bold Oblique
<b>*TIMES</b>	Times Roman
<b>*TIMESB</b>	Times Roman Bold
<b>*TIMESI</b>	Times Roman Italic
<b>*TIMESBI</b>	Times Roman Bold Italic
<b>*SYMBOL</b>	Symbol
<b>*DINGBATS</b>	Zapf Dingbats

You must also specify a font size in points. The default is 10-point.

**Example:** *CVTSPLSTMF FROMFILE(SALES)...  
TOFMT(\*PDF)  
FONT(\*CONVERT)  
CVTFNTRSC(ARIAL12 \*HELVETICA 12)*

The sales report is converted to PDF format. CVTSPLSTMF will attempt to map fonts, but the font named ARIAL12 is explicitly mapped to Helvetica 12-point.

## **BLANKS**

The **BLANKS** parameter only appears if **TOFMT(\*HTML)**, **TOFMT(\*HTXT)** or **TOFMT(\*TEXT)** is selected. This parameter allows you to define whether blank lines in the original report should be duplicated in the output.

<b>*NO</b>	No. Blank lines in the original report are not reflected in the output.
<b>*YES</b>	Yes. Blank lines in the original report are reflected in the output. Pages are padded out with blank lines to resemble the printed page.
<b>*FF</b>	Blank lines with form feeds. Blank lines in the original report are reflected in the output. At the end of each page, a form feed character is embedded in the output to force a page throw. This option is valid only in conjunction with TOFMT(*TEXT).

## **BOOKMARK**

The **BOOKMARK** parameter only appears if **TOFMT(\*PDF)** is selected. This parameter allows you to define whether PDF bookmarks (outlines) will be created, and, if so, on what basis.

Bookmarks index a PDF file so readers can go directly to a particular section of a document. By creating bookmarks, you make it quicker and easier for users to navigate the document that CVTSPLSTMF generates.

The possible values are:

<b>*PAGNBR</b>	CVTSPLSTMF creates bookmarks based on the page numbers in the document. This is the default.
<b>*NONE</b>	No bookmarks are created.
<b>*POS</b>	CVTSPLSTMF creates bookmarks based on a piece of text that appears at a particular position on each page of the report.
<b>*KEY</b>	CVTSPLSTMF creates bookmarks based on a text associated with a key word or phrase found in the report.

**Example:** *CVTSPLSTMF FROMFILE(SALES)...  
TOFMT(\*PDF)  
BOOKMARK(\*PAGNBR)*

The sales report is converted to PDF format and bookmarks are generated for each page of the report, labeled 'Page 1', 'Page 2' etc.

### **BMARKPOS**

The **BMARKPOS** (Bookmark String Position) parameter only appears if **BOOKMARK(\*POS)** is selected. This parameter allows you to define the position in the report where the text to be used as a bookmark regularly appears.

The parameter consists of 4 elements:

- I. the line number or y coordinate
- II. the column number or x coordinate
- III. the length of the text to be selected
- IV. the unit of measurement

By specifying **BOOKMARK(\*POS)** with the **BMARKPOS** parameter, you can create a set of bookmarks based on a piece of text that appears at a particular position on each page of the report.

For example, if you know that an item of user interest - such as an order number, a customer name, or a product code - appears regularly at position 3 of line 4 on every page and is up to 20 characters long, you can generate your bookmarks by specifying:

**Example: CVTSPLSTMF FROMFILE(SALES)...  
BOOKMARK(\*PAGNBR)  
BOOKMARK(\*POS)  
BMARKPOS(4 3 20 \*ROWCOL)**

The BMARKPOS parameter operates slightly differently depending on whether your spooled file is a traditional row/column-based report using non-proportional fonts, or a more sophisticated AFP report using non-proportional fonts. With non-proportional fonts, row and column positioning is largely meaningless, and the position of text on the page must be specified in terms of x- and y-coordinates.

If your report is a simple AS/400 report with just non-proportional fonts, select **\*ROWCOL** (Rows and Columns) as the unit of measure method, and specify the line number on which the text appears, the column at which it starts, plus its length.

If your report uses proportional fonts, specify **\*INCH** or **\*MM** as the unit of measure, and specify the number of inches or millimeters down the page (y coordinate) or across the page from left to right (x coordinate) the text appears, plus its length in inches or millimeters. It is also permissible to use **\*ROWCOL** for an AFP spooled file: CVTSPLSTMF will attempt to interpret the data in the AFP spooled file in terms of rows and columns, but be warned that the results may only be approximate.

**Example: CVTSPLSTMF FROMFILE(SALES)...  
TOFMT(\*PDF)  
BOOKMARK(\*POS)  
BMARKPOS(3 20 40 \*ROWCOL )**

Here the sales report is converted to PDF. Assuming that the spooled file is a simple SCS spooled file using only non-proportional fonts, and that the salesperson's name appears at position 20 on line 3 of each page, and is up to 40 characters long, this command will generate PDF bookmarks based on the name of the salesperson whose figures appear on each page.

**Example: CVTSPLSTMF FROMFILE(INVOICES)...  
TOFMT(\*PDF)  
BOOKMARK(\*POS)  
BMARKPOS(20 30 100 \*MM )**

In this example, the invoices spooled file is converted to PDF. Assuming that the spooled file is an AFP spooled file using proportional fonts, and that the customer's name appears at 20 mm down each page and 20 mm across each page, and is up to 100 mm long, this command will generate PDF bookmarks based on the name of the customer whose data appears on each page.

## **BMARKKEY**

The **BMARKKEY** (PDF Bookmark Key String) parameter only appears if **BOOKMARK(\*KEY)** is selected. This parameter allows you to define a key string

which will be used to identify the position in the report of the text to be used as a bookmark.

The parameter consists of 5 elements:

- the key string
- the occurrence of this string on the page to be used
- the offset from the start of the key string to the start of the bookmark text
- the length of the text to be selected
- the unit of measurement

By specifying **BOOKMARK(\*KEY)** with the **BMARKKEY** parameter, you can create a set of bookmarks based on a piece of text that appears in the report on the same line as another piece of text (i.e. the key string, e.g. a field label) .

For example, if you know that an item of user interest - such as an order number, a customer name, or a product code - appears on the same line as a label such as 'Customer name:', 'Order No:' or 'Product Id', then by using these labels as a key string you can generate bookmarks based on the actual customer name, order number or product code, e.g.

**CVTSPLSTMF ... BOOKMARK(\*KEY)**  
**BMARKKEY('Customer name:' 1 16 40 \*ROWCOL)**

The BMARKKEY parameter operates slightly differently depending on whether your spooled file is a traditional row/column-based report using non-proportional fonts, or a more sophisticated AFP report using non-proportional fonts. With non-proportional fonts, row and column positioning is largely meaningless, and the position of text on the page must be specified in terms of x- and y-coordinates.

If your report is a simple AS/400 report with fixed non-proportional fonts, select **\*ROWCOL** (Rows and Columns) as the unit of measure method, and specify the offset and length in characters.

If your report is uses proportional fonts, specify **\*INCH** or **\*MM** as the unit of measure, and specify offset and length in inches or millimeters respectively. It is also permissible to use **\*ROWCOL** for an AFP spooled file: CVTSPLSTMF will attempt to interpret the data in the AFP spooled file in terms of characters, but be warned that the results may only be approximate.

Note that the offset value is measured from the start of the key string. A negative offset indicates that the text to be selected occurs to the left of the key string, a positive offset indicates that it appears to the right.

The "occurrence" part of this parameter controls which occurrence of the key string on the page causes the bookmark text to be selected. That is, if you specify:

**CVTSPLSTMF ... BOOKMARK(\*KEY)**  
**BMARKKEY('Product code:' 1 30 100 \*MM)**

the bookmark will be generated from the text 30 millimeters to the right of the *first* occurrence of the string 'Product code:' on each page, whereas if you specify:

**CVTSPLSTMF ... BOOKMARK(\*KEY)  
BMARKKEY('Product code:' 2 30 100 \*MM)**

the bookmark will be generated from the text 30 millimeters to the right of the *second* occurrence of the string 'Product code:' on each page.

**Example: CVTSPLSTMF FROMFILE(SALES)...  
TOFMT(\*PDF)  
BOOKMARK(\*KEY)  
BMARKKEY('Area:' 1 7 30 \*ROWCOL )**

Here the sales report is converted to PDF. Assuming that the spooled file is a simple SCS spooled file using only non-proportional fonts, and that the sales area name starts 7 characters to the right of the label 'Area:' on each page, and is up to 30 characters long, this command will generate PDF bookmarks based on the first area on each page.

**Example: CVTSPLSTMF FROMFILE(INVOICES)...  
TOFMT(\*PDF)  
BOOKMARK(\*POS)  
BMARKKEY('20 30 100 \*MM )**

## **SPLIT**

The **SPLIT** (Split Spooled File) parameter only appears if **TOFMT(\*PDF)** is selected. This parameter allows you to request that **CVTSPLSTMF** create several stream files from a single spooled file, splitting the spooled file based on criteria that you specify on the **SPLITPAGE** and **SPLITKEY** parameters.

This option can be useful if your program produces a single spooled file which has traditionally been split up into separate pages or groups of pages prior to distribution on paper. You can ask **CVTSPLSTMF** to split the spooled file every so many pages, or when a specified key string appears in the spooled file.

**CVTSPLSTMF** generates names for the stream files it creates by appending a sequential number to the part of the stream file name specified on the **TOSTMF** parameter preceding any file extension. For example, if you specify:

**TOSTMF(pool.pdf)... SPLIT(\*PAGNBR) or ... SPLIT(\*KEY)**

**CVTSPLSTMF** will create stream files called **spool1.pdf**, **spool2.pdf**, **spool3.pdf** etc.

Options are:

- |              |  |
|--------------|--|
| <b>*NONE</b> | Do not split the spooled file. A single stream file is created.  |
| <b>*PAGE</b> | Split the spooled file into separate stream files every so many pages. This option is useful if, for example, you want to create separate stream files for each customer in the report, and the report starts each customer on a new page. |

**\*KEY** Split the spooled file into separate stream files every time a given key string appears in the spooled file. This option can be useful if, for example, you wish to split the report every time a piece of text (e.g. a field label such as 'Customer number:') appears in the report.

Note that each stream file must consist of at least one complete page. CVTSPLSTMF cannot split a single page between stream files.

### ***SPLITPAGE***

The **SPLITPAGE** (Split File Every n Pages) parameter only appears if **TOFMT(\*PDF)** and **SPLIT(\*PAGE)** are selected. This parameter allows you to specify the number of pages after which **CVTSPLSTMF** will create a new stream file.

**Example:** **CVTSPLSTMF FROMFILE(INVOICES)...  
TOFMT(\*PDF)  
SPLIT(\*PAGE)  
SPLITPAGE(1)**

Here the invoices spooled file is split into separate stream files for each page of the report.

### ***SPLITKEY***

The **SPLITKEY** (Split Key String) parameter only appears if **TOFMT(\*PDF)** and **SPLIT(\*KEY)** are selected. This parameter allows you to specify a key string which will trigger the creation of a new stream file every time it appears in the spooled file.

**Example:** **CVTSPLSTMF FROMFILE(SALES)...  
TOFMT(\*PDF)  
SPLIT(\*KEY)  
SPLITKEY('Branch code:')**

Here the sales report is split into separate stream files every time the string 'Branch code:' appears in the report.

Note that the stream file that is created when the key string is detected in the spooled file will start at the beginning of the page on which the key string first appears and will continue until the page preceding the next occurrence of the key string in the spooled file. However, if the key string appears more than once per page, occurrences following the first occurrence are ignored. In other words, a stream file always consists of at least one complete page, irrespective of how many times the key string appears per page. CVTSPLSTMF cannot create a PDF file that represents only a part of a single page of the report. If this limitation is an inconvenience, you should consider adding a page throw prior to each occurrence of the key string in the report. Remember: with CVTSPLSTMF, paper costs are no longer an issue!

## **EXITPGM**

The **EXITPGM** (Exit Program) parameter only appears if **TOFMT(\*PDF)** is selected. This parameter allows you to specify the fully qualified name of a user-written exit program which CVTSPLSTMF will call after completing each stream that it creates.

Specify **\*NONE** if you do not wish to call an exit program, or specify the name of the program you wish to call.

Specify the program name and the library in which the program is located. Alternatively, specify one of these special values for the library name:

- \*LIBL**            The program is located using the library list of the job.
- \*CURLIB**        The program is located in the current library.

The exit program could be used for a variety of purposes. Some of these are explored below.

Firstly, you might split your invoices spooled file into separate files for each invoice using the **SPLIT(\*PAGE)** option, like this:

**Example:    CVTSPLSTMF    FROMFILE(INVOICES)  
                              TOSTMF(invoice.pdf)...  
                              TOFMT(\*PDF)  
                              SPLIT(\*PAGE)  
                              SPLITPAGE(1)**

CVTSPLSTMF will create spooled files called **invoice1.pdf**, **invoice2.pdf**, **invoice3.pdf** etc. You could use the exit program to rename these files to something more meaningful. For example, using the parameter information passed to the exit program you might rename these files to something based on your customer number or the customer name.

Secondly, the exit program could be used to perform further processing on the stream file. For example, the exit program could use the parameter information it is passed to identify the customer to whom the invoice relates, retrieve the customer's e-mail address from the database, and invoke a mail application to send the invoice to the customer.

See the **EXITPGMPRM**, **EXITPGMPOS** and **EXITPGMKEY** parameters for further information about how to specify what parameter information should be passed to your exit program.

See Appendix A for an example of an exit program which renames the stream file just created based on the parameter data it is passed.

The exit program is passed a standard set of parameters, described in the table below:

1	Spooled file name	Input	Char(10)
---	-------------------	-------	----------

2	Name of the job that created the spooled file	Input	Char(10)
3	User id that created the spooled file	Input	Char(10)
4	Number of the job that created the spooled file	Input	Char(6)
5	Spooled file number	Input	Binary(4)
6	Spooled file user data	Input	Char(10)
7	Name of the stream file just created	Input	Char(128)
8	Directory name in which stream file was created	Input	Char(256)
9	User-definable exit program parameter	Input	Char(1024)

The user-definable exit program parameter is the text extracted from the report according to the criteria specified on the **EXITPGMKEY** or **EXITPGMPOS** parameters.

### **EXITPGMPRM**

The **EXITPGMPRM** (Exit Program) parameter only appears if **TOFMT(\*PDF)** is selected and you specify a program name on the **EXITPGM** parameter. This parameter allows you to define whether CVTSPLSTMF should pass any data to the exit program in the user-definable exit program parameter, and, if so, on what basis.

Refer to the **EXITPGM** parameter above for further information about the parameters passed to an exit program.

Options:

- \*NONE** No text will be extracted from the report. The user-definable parameter will consist of all blanks.
- \*POS** CVTSPLSTMF will pass a parameter consisting of an item of text extracted from the spooled file as defined by criteria specified on the **EXITPGMPOS** parameter.
- \*KEY** CVTSPLSTMF will pass a parameter consisting of an item of text extracted from the spooled file as defined by criteria specified on the **EXITPGMKEY** parameter.

### **EXITPGMPOS**

The **EXITPGMPOS** (Exit program parameter string position) parameter only appears if **EXITPGMPRM(\*POS)** is selected. This parameter allows you to define the position in the report from which CVTSPLSTMF will extract an item of text



page number of 1 on this parameter would refer to the first page in the two stream files, i.e. pages 1 and 6 from the original spooled file.

- The key string itself. This could be a field label, e.g. 'Customer name:'
- The occurrence of the key string which will trigger the extraction of parameter text. For example, a value of 1 on this parameter will extract parameter text the first time the key string is located, a value of 2 on this parameter will extract parameter text the second time the key string is located etc.
- The offset from the start of the key string to the start of the text to extract for use as a parameter. This value can be negative if the text to extract is located to the left of the key string. This offset is measured in characters if unit of measure **\*ROWCOL** is specified, in inches if **\*INCH** is specified or in millimeters if **\*MM**.
- The length of the text to extract. This length is measured in characters if unit of measure **\*ROWCOL** is specified, in inches if **\*INCH** is specified or in millimeters if **\*MM**.
- The unit of measure. If your report is a simple AS/400 report with non-proportional fonts, specify **\*ROWCOL** as the measurement method. If your report is uses proportional fonts, specify **\*INCH** or **\*MM** as the measurement method.

```
Example:  CVTSPLSTMF  FROMFILE(INVOICES)
           TOSTMF(invoice.pdf)...
           TOFMT(*PDF)
           SPLIT(*PAGE)
           SPLITPAGE(1)
           EXITPGM(Invexit)
           EXITPGMPRM(*KEY)
           EXITPGMKEY(  1
                       'Customer name:'
                       1
                       15
                       50
                       *ROWCOL)
```

In this example, the invoices spooled file is converted to separate PDF files for each page of the report. Every time a file has been completed, a program called INVEXIT will be called. The program will be passed 50 characters of text extracted starting 15 characters to the right of the first occurrence of the string 'Customer name:' on the first page written to each stream file.

## **DELIMITERS**

The **DELIMITERS** (Delimiters for \*CSV) parameter only appears if **TOFMT(\*CSV)** is selected. This parameter allows you to control several aspects of the process of creating a delimited ASCII file from your spooled file.

There are 5 elements to the parameter.

### **Record delimiter**

This element allows you to specify the characters to be used to indicate the end of a record in the CSV file.

Options are:

<b>*CRLF</b>	Carriage return and line feed. Both a carriage return (x'0D') and a line feed (x'0A') character are used.
<b>*CR</b>	Just a carriage return (x'0D') is used.
<b>*LF</b>	Just a line feed (x'0A') is used.

### String delimiter

This element allows you to define the character that encloses string (alphanumeric) data in the delimited ASCII file that is to be created.

Either type the character to be used, or select one of the special values:

<b>*DBLQUOTE</b>	A double quote (") is used
<b>*SGLQUOTE</b>	A single quote (') is used
<b>*NONE</b>	No delimiter is used. Alphanumeric data is not enclosed by any special character.

### Field Delimiter

This element allows you to define the character that separates fields in the delimited ASCII file that is to be created.

Either type the character to be used, or select one of the special values:

<b>*COMMA</b>	A comma (,) is used
<b>*TAB</b>	A tab (x'09') is used
<b>*BLANK</b>	A blank (x'20') is used
<b>*SEMICOLON</b>	A semicolon (;) is used.

**Example:** *CVTSPLSTMF FROMFILE(SALESSTATS)...  
TOFMT(\*CSV)  
DELIMITERS(\*CRLF \*DBLQUOTE \*COMMA)*

The Sales Stats report is converted to a delimited file in CSV (Comma-separated variable format). Records are terminated by a carriage return/line feed pair. Alphanumeric data is enclosed in double quotes. Fields are separated by commas.

### Column Separator

The values you select on this parameter control how the spooled file is converted to a delimited ASCII text file (e.g. a comma separated variable file).

There are two elements to this parameter: the column separator character and the number of consecutive separator characters to force a column break.

The default for the first element is \*SPLF. This tells CVTSPLSTMF to break the spooled file data into columns based on the way in which the data is split up in the spooled file.

Where the spooled file is created from an externally described printer file this will probably give the best results, since the data in the spooled file is likely to be organized so that each natural or logical item of data appears as a separately identifiable element.

However, if the spooled file is created from an internally described printer file, or from an application such as Query/400, it is likely that the data in the spooled file will be presented to CoolSpools as a single, unstructured data block for each line of the report. If this is the case then better results will probably be achieved by defining a column separator character to control the splitting of the data in the report.

The column separator character can be any alphanumeric character. Often the best results will be obtained by selecting \*BLANK (blank character) which will split the data in the spooled file every time one or more consecutive blank characters occur.

Options are:

<b>*SPLF</b>	Split the data in the report based on the organization of the data
<b>*BLANK</b>	The separator character is the blank (space) character.
<b>Character</b>	The separator character to use to identify column breaks.

### **Number of column separators**

The number of consecutive column separator character that must appear before a column break occurs.

## ***PDFVIEWER***

The **PDFVIEWER** (PDF Viewer Controls) parameter allows you to define the type of PDF viewer application that will be used to read the PDF file to be created.

This parameter is ignored unless TOFMT(\*PDF) is also specified.

There are 4 elements to this parameter.

### **PDF viewer type**

When creating a PDF file, this element controls the way in which fonts are defined. If you intend to use a viewer other than a Microsoft ® Windows viewer application, for example the Linux version of Adobe ® Acrobat, specify VIEWER(\*OTHER) on this parameter.

Unless VIEWER(\*OTHER) is specified, the PDF file may not appear correctly when viewed using the Linux version of Acrobat or any other PDF viewer that does not run under Windows.

Options are:

- \*WINDOWS** A Microsoft ® Windows PDF viewer will be used
- \* OTHER** A viewer other than a Microsoft ® Windows viewer will be used.

### **Initial bookmark action**

This element allows you to specify whether PDF bookmarks should be visible when the report is first opened.

Options are:

- \*SHOW** (The default) Show the bookmarks when the document is opened
- \*HIDE** Do not show the bookmarks when the document is opened.

It should be noted that some improvement in the time taken to open a PDF file can be accrued if BMARKACT(\*HIDE) is selected. However, your users will need to make the bookmarks visible before they can be used to navigate around the document.

**Example: CVTSPLSTMF FROMFILE(SALES)...  
BOOKMARK(\*PAGNBR)  
BMARKACT(\*HIDE)**

The sales report is converted to PDF format and bookmarks are generated for each page of the report. However, the bookmarks are not displayed when the PDF file is first opened.

### **Initial zoom when PDF opened**

This element allows you to define an initial magnification to be used when a PDF file is first opened.

This parameter is ignored unless TOFMT(\*PDF) is also specified. This parameter controls the appearance of the PDF document when first opened. Specifically, it determines the magnification factor used to display the document in the PDF viewer application.

Options are:

- \*PDFDFT** The default magnification defined in your viewer options is used.
- \*FITWDW** Adjust the magnification so that the entire page just fits in the viewer window.
- \*FITWIDTH** Adjust the magnification so that the width of the page just fits in the viewer window.

<b>*FITVIS</b>	Adjust the magnification so that the text and graphics on the page fit in the viewer window.
<b>*ACTUAL</b>	Adjust the magnification so that the page is viewed at its actual size.
<b>Zoom factor</b>	The percentage magnification to apply.

### **PDF keywords for indexing**

This element allows you to define a set of keywords to be included in the Document Info section of the PDF file. These can be used by indexing and document management applications. Specify the keywords as a single character string with keywords separated by a comma or semicolon.

### ***EXCLLINBR***

The **EXCLLINBR** (Exclude Line Numbers) parameter specifies the line numbers of lines on the report which should be excluded from the Excel spreadsheet.

This parameter is ignored unless TOFMT(\*XLS) is also specified.

There are two elements to this parameter.

- The starting line number of the set of lines to be excluded
- The number of lines to exclude from this starting line number onwards

Up to 20 sets of lines can be specified on this parameter.

### ***EXCLLINKEY***

The **EXCLLINKEY** (Exclude Lines by Key) parameter specifies sets of lines on the report which should be excluded from the Excel spreadsheet based on the appearance in the line of a key string .

This parameter is ignored unless TOFMT(\*XLS) is also specified.

There are two elements to this parameter.

- The key string which identifies the start of a set of lines to be excluded
- The number of lines to exclude from this starting line onwards

Up to 20 sets of lines can be specified on this parameter.

### ***WORKSIZE***

The **WORKSIZE** (Initial Work File Member Size) parameter allows you to control the initial and maximum size of the work file used by **CVTSPLSTMF**.

When converting very large spooled files (e.g. in excess of 2,000-3,000 pages), the CVTSPLSTMF work file may reach its default maximum size and require you to respond to increment messages in the QSYSOPR message queue. This can be avoided by specifying a larger maximum size or \*NOMAX on this parameter.

There are three elements to this parameter.

- The initial number of records that the member can contain before increments are added (\*NOMAX may be specified to indicate that there is no maximum)
- The number of records in each increment
- The maximum number of increments allowed

## **TODIR**

The **TODIR** (To Directory) parameter specifies the directory path where the stream file should be saved.

There are two elements to this parameter.

### **1. To directory**

The default option is \***TOSTMF**, which indicates that the full directory path to the stream file has been specified on the **TOSTMF** parameter and that this parameter is unused.

If you specify a directory name here it must already exist, otherwise an error will occur and the file will not be saved.

Think of a path name as being a little like the address on a letter, only in reverse. When you write an address on an envelope, you start with the number or name of the house or building, then the street name, the city, the state, and finally the country, e.g.

**The White House  
Pennsylvania Avenue  
Washington  
DC  
U.S.A.**

Here we are moving from the smallest entity (building) to the biggest (country). However, the postal services will read the address in the reverse order, deciding which country to send the letter to first before looking for the state and so on.

Like the Post Office, the AS/400 will examine a path name starting with the top-level entity (e.g. a directory in the root of the IFS) and working down toward lower level entities (sub-directories and individual files), e.g. a Company which has many branches worldwide might organize its sales figures files in a directory structure something like this:

**/usa/western/ca/sandiego/mainstreet/sales.csv**

Path names can be either **absolute** or **relative**.

An absolute path name is one which starts with a slash character (either forward or backward, / or \), e.g.

**/sales/2000/october/stats/sales.csv**

Since this path name starts with a slash, the AS/400 will interpret this path name as starting in the **root** of the IFS (Integrated File System), i.e. at the highest level, and moving down through a directory called **sales** to a sub-directory called **2000**, further sub-directories called **october** and **stats**, and finally to a file called **sales.csv**. This is a little like the full postal address of the file, with **sales** being equivalent to the country, **2000** to the state, **october** to the city, **stats** to the street and **sales.csv** to the actual building where the letter is eventually delivered.

Alternatively, one can use a **relative** path name. This is like a partial postal address which assumes a starting point. For example, if you're in London and someone asks you the way to Birmingham, you're unlikely to enquire "Birmingham, England or Birmingham, Alabama"? You can assume from the context of the question that the Birmingham in England is the one being referred to.

Similarly, if you specify a path name without a starting slash symbol, the AS/400 will assume a starting point for this address, namely your **current directory**. Each time you sign on to the AS/400, your current directory will be set automatically to the value which your Security Officer specified on the **HOMEDIR** parameter of the CRTUSRPRF (Create User Profile) command when your user profile was first created. You can change your current directory at any time subsequently by means of the **CHGCURDIR** (Change Current Directory) command (for which the short form **CD** exists as an alternative). For example,

**CHGCURDIR /sales**

or **CD /sales**

Assuming that your current directory has been set to **/sales** like this, then the relative path name

**2000/october/stats/sales.csv**

and the absolute path name

**/sales/2000/october/stats/sales.csv**

both refer to the same file.

If you have users who still use shared folders, you can refer to the shared folder file system by starting your path name with **QDLS**, e.g.

**/qdls/sales/2000/october/sales.csv**

This path name refers to a hierarchy of shared folders starting in a folder called **sales**.

**Example: CVTSPLSTMF FROMFILE(QSYSPRT)  
 TOSTMF(sales.pdf)  
 TODIR('/sales/2001/april')**

Here the CVTSPLSTMF command is being applied to a spooled file called **QSYSPRT** in order to create a stream file called **sales.pdf** in an IFS root directory called **sales/2001/april**.

## 2. Directory for PDF work files

The second element to the TODIR parameter allows you to specify the directory in which work files are created when generating a PDF file.

The default is \*TODIR, which indicates that the directory where the PDF file will be created is used. Alternatively, you may wish to specify a different directory, e.g. TMP.

## STMFCODPAG

The **STMFCODPAG** (Stream File Code Page) parameter allows you to indicate the code page to which data in the spooled file should be converted. The code page controls how character data in the spooled file is translated from its AS/400 EBCDIC representation to the ASCII format needed by PC applications.

You may use one of the special values:

- \*PCASCII** The contents of the spooled file are translated from EBCDIC to a Windows-compatible version of ASCII based on the CCSID of the current job and the code page table shown below.
- \*STMF** The contents of the spooled file are translated from EBCDIC to Windows-compatible version of ASCII based on the CCSID of the current job and the code page of the existing stream file. If the stream file does not exist, this parameter is ignored and a CCSID is selected as per \*PCASCII above.

Alternatively, you may enter a specific code page number.

Common Windows code pages are shown in the table below. CVTSPLSTMF maps EBCDIC CCSIDs to Windows code pages on this basis. If the EBCDIC CCSID has no mapping defined, code page 1250 is used by default. If the characters in your spooled file do not appear correctly in the stream file created by CVTSPLSTMF, try specifying one of the following values on the STMFCODPAG parameter.

Code Page	Code Page Name	Example countries where this Code Page is appropriate
1250	Windows Latin-2	US, Canada, Netherlands, Portugal, Brazil, New Zealand, Australia, Iceland

1251	Windows Cyrillic	Russia, Serbia, Bulgaria
1252	Window Latin-1	UK, France, Italy, Germany, Austria, Benelux, Canada, Switzerland, Spain, Latin America, other European
1253	Windows Greek	Greece
1254	Windows Turkish	Turkey
1255	Windows Hebrew	Israel
1256	Windows Arabic	Arabic-speaking countries
1257	Windows Baltic Rim	Denmark, Norway, Finland, Sweden, Estonia

**Example:** *CVTSPLSTMF FROMFILE(FACTURES)...  
STMFCODPAG(1252)*

Here the CVTSPLSTMF command is being applied to a French-language spooled file called **FACTURES**. Code page 1252, suitable for French-language data, will be used to convert the contents of the spooled file.

## **AUT**

The **AUT** (Public data authorities) parameter allows you to control the public data authorities given to stream files created as part of the CVTSPLSTMF command.

There are two elements to this parameter.

### **1. Public data authority**

This element allows you to define the public authority for the stream to be created. If the stream file already exists, this parameter has no effect and the existing stream file retains its current authority settings.

The owner of the file is always granted full authority. This parameter controls the authority given to other users.

The default is **\*R** (read-only authority).

Options are:

<b>*R</b>	Read only
<b>*W</b>	Write only
<b>*X</b>	Execute only
<b>*RW</b>	Read and write
<b>*RX</b>	Read and execute
<b>*WX</b>	Write and execute
<b>*RWX</b>	Read, write and execute (all)
<b>*NONE</b>	No authority

**CVTSPLSTMF** needs to create three temporary work files when it is generating PDF output. These files are normally automatically deleted when the command completes or if any error occurs. The files are named **#job\_nbrn**, where **job\_nbr** is the job number of the job in which the command is run and **n** is the work file number (1, 2, 3).

The default is **\*TODIR** (To Directory), which indicates that the work files will be created in the same directory as the final PDF file (as specified on the **TOSTMF** or **TODIR** parameters).

This parameter gives you the option to create work files in a different directory from the directory where the final output will be created. For example, you might use a “temp” directory. This can be useful if you wish to avoid users seeing the temporary files while PDF files are being created.

## 2. Authority for PDF work files

The second element of the AUT parameter allows you to define the public authority for the work files used when creating a PDF file.

**CVTSPLSTMF** needs to create three temporary work files when it is generating PDF output. These files are normally automatically deleted when the command completes or if any error occurs. The files are named **#job\_nbr**, where **job\_nbr** is the job number of the job in which the command is run.

The owner of the temporary file is always granted full authority. This parameter controls the authority given to other users.

The default is **\*RWX** (read, write and execute authority, i.e. full authority). Please note that if a value other than **\*RWX** is specified on this command, it is possible in certain circumstances for the creator of the file not to be authorized to delete it subsequently, and **CVTSPLSTMF** may be unable to clear them up after use.

Options are:

<b>*R</b>	Read only
<b>*W</b>	Write only
<b>*X</b>	Execute only
<b>*RW</b>	Read and write
<b>*RX</b>	Read and execute
<b>*WX</b>	Write and execute
<b>*RWX</b>	Read, write and execute (all)
<b>*NONE</b>	No authority





## Appendix A: Sample Exit Program

This program is an example of the way in which the CVTSPLSTMF exit program facility can be used. In this example, the exit program is called to rename the stream file created by the command to something more meaningful. It is assumed that the EXITPGMPRM(\*POS) or EXITPGMPRM(\*KEY) option has been used to extract an item of information from the report so that it is passed to this exit program in the ninth parameter position. This could be, for example, a customer number, order number, invoice number etc. The stream file is renamed from the name given it by CVTSPLSTMF to nnnnnnn.pdf, where nnnnnnn is the number extracted from the report and passed as a parameter.

```
*****
*
* PROGRAM NAME: EXITPGM
*
* LANGUAGE      : ILE RPG
*
* DESCRIPTION   : Sample CVTSPLSTMF exit program
*
*****

H COPYRIGHT(' (C) @riadne software July 2000')

*****
* PROTOTYPE FOR STANDARD EXIT PROGRAM PARAMETERS
*****

D Parameters      PR              ExtPgm('EXITPGM')

* Spooled file name

D   iaSplFile          10A

* Name of the job which created the spooled file

D   iaSplJob           10A

* User id of the job which created the spooled file

D   iaSplUser          10A

* Job number of the job which created the spooled file

D   iaSplJobNo         6A

* Spooled file number of the spooled file

D   iiSplNbr           10I 0

* Spooled file user data

D   iaUsrDta           10A

* Name of the stream file created

D   iaStmFile          128A
```

```

* IFS directory in which stream file was created
D   iaDir                256A

* Exit program parameter string extracted from spooled file
D   iaExitParm           1024A

*****
* PROTOTYPE FOR QCMDEXC
*****

D Command                PR                ExtPgm('QCMDEXC')
D   iaCommand            32767A          CONST OPTIONS(*VARSIZE)
D   inCmdLen             15P 5          CONST

*****
* STANDARD EXIT PROGRAM PARAMETERS
*****

D Parameters            PI
D   iaSplFile           10A
D   iaSplJob            10A
D   iaSplUser           10A
D   iaSplJobNo          6A
D   iiSplNbr            10I 0
D   iaUsrDta            10A
D   iaStmFile           128A
D   iaDir                256A
D   iaExitParm          1024A

D laOldName             S                1024A
D laNewName             S                1024A

D QUOTE                 C                ''''

* If customer number found and passed as parameter, rename stream file
C                        IF                iaExitParm <> *BLANKS

* Set up full path name to stream file
C                        EVAL                laOldName = QUOTE
C                        + %trim(iaDir)
C                        + '/'
C                        + %trim(iaStmFile)
C                        + QUOTE

* Set up new path name for file using customer number in parameter
C                        EVAL                laNewName = QUOTE
C                        + %trim(%subst(iaExitParm:1:7) )
C                        + '.pdf'
C                        + QUOTE

* Rename the stream file
C                        CALLP(E) Command( 'REN '
C                        + laOldName
C                        + ' '
C                        + laNewName:2053)

C                        ENDIF

```

C

RETURN

Please contact @riadne software by e-mail to [support@ariadnesoftware.co.uk](mailto:support@ariadnesoftware.co.uk) if you require further assistance with programming exit programs or if you have any other questions, problems or suggestions relating to **CoolSpools**. We can provide other sample exit programs explaining, for example, how to copy PDF files to a server using FTP or e-mail them to your customers automatically.