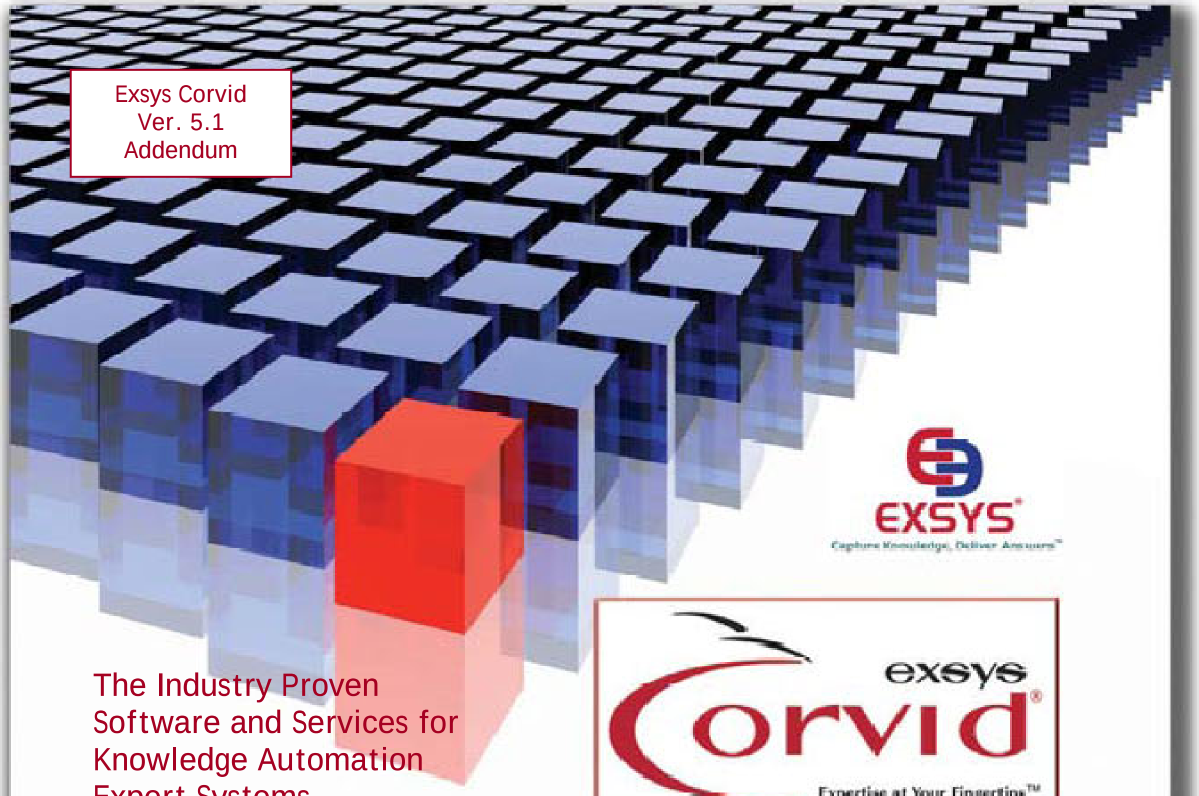


Exsys Corvid
Ver. 5.1
Addendum

EXSYS
Capture Knowledge, Deliver Answers™

exsys Corvid®
Expertise at Your Fingertips™

The Industry Proven
Software and Services for
Knowledge Automation
Expert Systems

Exsys Corvid Knowledge Automation Expert System Development Manual

V5.1 Addendum

Any references to company names in samples or exercises are for demonstration purposes only and are not intended to refer to any actual organization.

Exsys Corvid v5.1 Addendum

Summary

Exsys Corvid v5.1 makes it easy to build reports in the PDF format. This allows building complex reports that your end user can view, print and save using the PDF standard. Corvid systems have always been able to create HTML and RTF reports, but the PDF format has been quite difficult to generate due the complex PDF syntax. Corvid v5.1 includes a special program that makes it easy to build PDF reports that can be incorporated into your systems. PDF reports can be created and displayed using either the Corvid Servlet Runtime or the Corvid Applet Runtime, though the applet runtime requires the support of server-side programs. New command builder options in the Corvid development environment make it easy to build the required commands.

The Corvid Servlet Runtime v5.1 also adds a very convenient new feature making it much easier to moving Corvid systems from one server to another, and a new property that can be used with Collection Variables.

Creating PDF reports

Often the conclusion of a Corvid system will be to display a customized report to the user.

PDF documents are a desirable format for the report because:

1. They allow very complex formatting, including matching legal documents
2. They are quite portable across operating systems and browsers
3. They are easy to print, save and distribute
4. They can not be edited by the end user.

PDF reports can better match the look of published reports and can use the capability of the expert systems to build and fill-in complex forms. The other common formats for reports include .txt, .html, .xml, and .rtf. While these work well in many cases, PDFs are best for complex and precise formatting of reports that are intended to be saved or printed.

Prior to Corvid v5.1, it was possible to build PDF reports with Corvid, but it was quite difficult and required a detailed knowledge of the very complex syntax of PDF documents. Corvid v5.1 adds new commands and server-side programs that make building PDF reports quite easy. The basic steps of creating a report is to put the contents of the report into a Collection variable, then use the special new versions of the SaveReport and DisplayReport commands to save the report on the server and display it to the user.

Corvid Server Programs Needed for Reports

To build any report that is displayed in a separate browser window you will need the CorvidRpt servlet installed on your server. If you plan to build PDF reports you will also need to install the ExsysFop servlet. This is an extension of the open source FOP program. The Exsys extensions change the way the program is called and where files are created on the server. The core FOP functionality is not changed. Since this is an open source program, the source code for the Exsys modifications is also provided.

Corvid v5.1 includes a new CorvidRpt servlet program, which should replace any earlier CorvidRpt program.

Install CorvidRpt on the Server

Deploy CorvidRpt.war on the server. For example for Tomcat, copy the file:

C:\Program Files\Exsys\Corvid\ServerPrograms\CorvidRpt.war

to Tomcat's 'webapps' folder. Use Tomcat's Manager page to deploy it or restart Tomcat. Create the CorvidReport.cfg file which specifies where the reports will be saved by the SaveReport command. Create it in NotePad and the first line is the full path to the folder where the reports will be saved. For example:

C:\reports

Put the CorvidReport.cfg file in Tomcat's 'webapps/CorvidRpt/WEB-INF' folder. If unable, see Troubleshooting below.

Install ExsysFop on the Server

FOP is an open source program created by Apache. It converts *.fo* files (which are XML files) into *.pdf* files which can be opened in a user's browser. This is used as one of the steps in building PDF reports. Exsys has created a customized version of FOP to make it easier to use with Corvid and to increase security. The .war file needs to be fielded on the server. The .war file is part of the Corvid v5.1 (and later) install and can be found in:

C:\ Program Files\Exsys\Corvid\ServerPrograms\ExsysFop.war

The customized version, ExsysFop, passes the passwords created by SaveReport and enforces the other security options.

Deploy the ExsysFop.war file the same way as the CorvidRpt.war file. After it is deployed, copy the CorvidReport.cfg file to the ExsysFop servlet's WEB-INF folder (e.g. Tomcat's 'webapps/ExsysFop/WEB-INF' folder). This tells ExsysFop where the saved report files are located.

Since FOP is an open source program, the source code for the Exsys modifications is available should you need it.

Changes in Corvid v5.1 Concerning Reports

Corvid v5.1 slightly changes the steps for creating and displaying all types of .txt, .xml, .rtf and .pdf reports that use the SaveReport/DisplayReport commands. This was necessary because Microsoft Internet Explorer 8 changed the way the browser recognizes the “type” of a document. The browser must correctly recognize the document type to know how to handle it and to know if it should be passed to another program. For example RTF documents are passed off to MS Word and PDF documents are (typically) passed off to Adobe Acrobat. While this only applies to IE8 and other browsers do not have the problem, it required modifying the format of the URL Corvid creates to display various types of documents so that they would work with all current browsers.

To the Browser, reports are "generated" by the CorvidRpt servlet even though it is really just echoing what the expert system sent it. The server tells the browser what type of document is being sent by including the "content-type" in the http header. Most browsers, including earlier versions of Internet Explorer either used the “content-type” or only required adding the appropriate extension in the URL. IE 8 requires an extension at a different location in the URL. To prevent such future changes in IE from affecting Corvid, CorvidRpt now uses URLs that are indistinguishable from the URL for a real document file (e.g. A normal PDF file on the server).

Corvid v5.1 makes specifying the type of report easy by simply selecting it from a drop down list when building the commands. This "Type" is then used to build URLs that all current browsers will correctly handle.

Corvid v5.1 also now allows reports to be saved in subfolders of the report folder on the server. If you have many expert systems, this feature can be used to organize each system’s reports into their individual folders. It can also be used to save the most important reports separate from the rest of the reports.

Backward Compatibility

The changes to Corvid v5.1 in report handling have been made backwardly compatible as much as possible, and very few older systems will require any change. Your system will only need to be modified if it uses the SaveReport/DisplayReport commands to build non-HTML reports using server-side programs or the Corvid Servlet Runtime. This is only likely to be an issue if your system builds and displays a RTF or XML report. If your system builds only HTML reports, or displays results using the Corvid Applet Runtime or a Results screens, no changes are required.

If your system builds RTF or XML reports, all you have to do is open the system with Corvid v5.1 and save it to build a new CVR file. Deploy the new CVR with the new Corvid v5.1 CorvidRpt.war to the server. For applets and applications, this is sufficient. For systems run using Corvid Servlet Runtime with non-html reports, see the instructions below. This will make your system compatible with all current browsers, including IE8 and any other browsers that use similar syntax.

Building PDF Reports

Building a PDF report has 5 main steps:

1. Formatting the report using RTF
2. Convert the RTF report template to an FO template
3. Putting the report in a Collection Variable
4. Saving the Report to the server
5. Displaying the report via a URL

1. Formatting the Report Using RTF

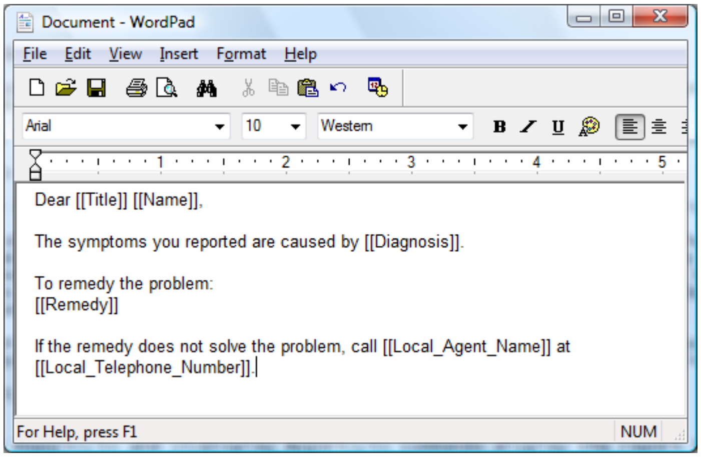
RTF (Rich Text Format) is used to design and build the structure of the report. The RTF will later be converted to PDF, but we must start with RTF because it has a much easier structure to read and work with. Also, RTF documents can be easily built with MS Word and many other text editing programs. RTF provides a wide range of formatting options allowing even complex reports and forms to be created.

RTF files are text. If you open one in Notepad (NOT WordPad or MS Word) you can see the text commands and syntax that makes up the RTF document. Because of this, RTF documents allow double square bracket embedding of variables that Corvid will then replace with content from the expert system. Also, sections of RTF code can be combined to build a larger document, though this requires some knowledge of RTF syntax.

PDF files on the other hand can not be easily opened and read. While PDF documents can also be directly created from MS Word and other programs, these can NOT be used directly with Corvid due to their complex internal structure.

The easiest way to build a RTF document is in MS Word or the WordPad program that comes with MS Windows. (Note: Do NOT use Notepad because it does not support RTF.) MS Word allows much more complex formatting and layout than WordPad.

The easiest way to build the content of the report is to use double square bracket embedded Corvid variables. Just add the name of the variable in [[ ]] at the appropriate places and Corvid will replace the [[..]] expression with the variable's actual value. For example, below is a very simple report built in WordPad.



Dear [[Title]] [[Name]],

The symptoms you reported are caused by [[Diagnosis]].

To remedy the problem:
[[Remedy]]

If the remedy does not solve the problem, call [[Local_Agent_Name]] at
[[Local_Telephone_Number]].

Once the report is built, select "Save As" from the "File" menu and select "Rich Text Format (RTF)" from the drop down at the bottom of the window. Give the file a name and save it with the files for your Corvid system.

The same approach can be used in MS Word, which provides many more formatting options for complex documents. Again, save the document as RTF.

Any layout and formatting options can be used to design the template. However, be very careful NOT to change formatting such as font, size, color, etc options in the [[..]] expressions. RTF syntax includes codes whenever the text format changes and if these appear in the [[..]], Corvid will not later be able to recognize the variable name. This will lead to a “Unrecognized variable” error later when running the system.

If you get a runtime error or just want to check the file, open the file in Notepad. Notepad does not support RTF and will display the RTF codes as text. This may be quite complex, but search the file for "[[" and make sure that the [[..]] expressions only contain the name of the Corvid variable and do not have any additional RTF codes. If they do have RTF codes in them, opening the document in Word or WordPad, selecting the full “[[..]]” expression and setting the format for it should get rid of the RTF codes in the variable name.

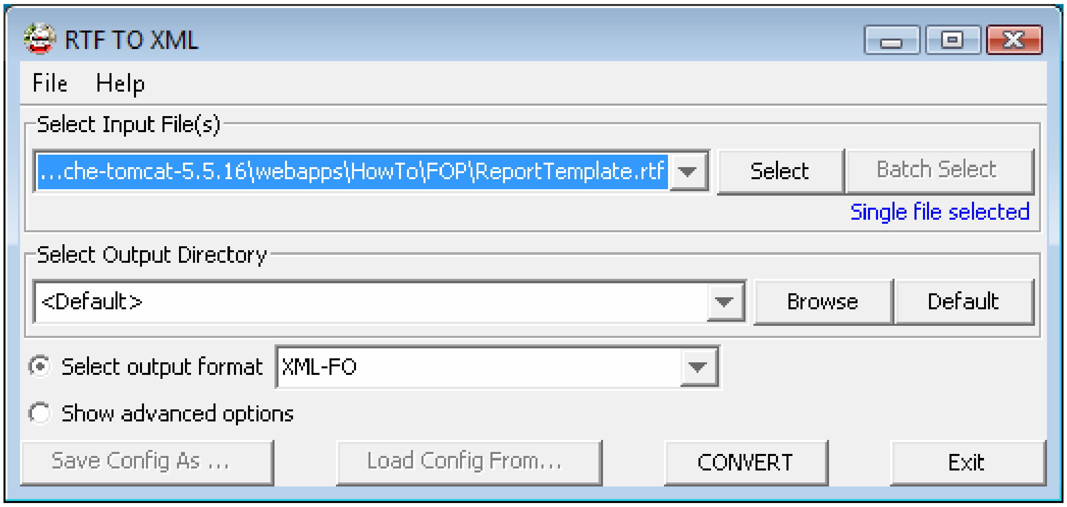
2. Convert the RTF Template to an FO Template

FO is a standard way to represent RTF as XML. This may seem like an odd thing to do, but the advantage of FO is that it can be converted to PDF.

There are various ways to create the .fo template file. One of the easiest is to use an inexpensive program called 'RTF_to_XML'. FO can also be created from a XHTML, XML, DocBook, or TEI document by using an XSLT transform to create the FO file. It can also be created directly, though this requires a good knowledge of FO syntax.

At the moment, the easiest way to build FO files is to use "RTF_to_XML", which is available from **http://www.rtf-to-xml.com**. This is a third party program and not part of the Corvid package. The current price is $40, and it can be ordered online. Since FO is a standard, it is expected that there will be more and more ways to directly build FO files via “Save As” options in text editing programs.

To convert the RTF template to a .fo template using RTF_to_XML, build the RTF template as described above. Start RTF_to_XML and select the .rtf file as the Input File and pull down XML-FO as the Output Format and click the Convert button to create the .fo template file, which you will find in the same folder.



RTF TO XML

Select Input File(s)

...che-tomcat-5.5.16\webapps\HowTo\FOP\ReportTemplate.rtf

Single file selected

Select Output Directory

<Default>

● Select output format XML-FO

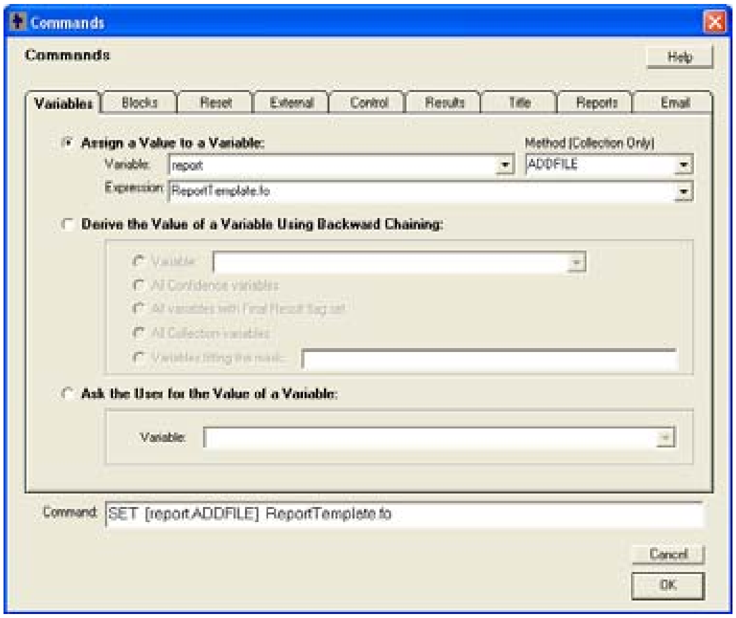
○ Show advanced options

This will produce an FO template that we can use to build a PDF file.

3. Putting the FO Report in a Collection Variable

The next step is to put the content of the FO template into a Corvid Collection variable. This collection variable must eventually contain the full report. It will be used with the SaveReport command to put the report on the server.

If the report is based on a single FO template with embedded [[..]] expressions this is very easy to do. Just create a Corvid Collection variable to hold the report and use the ADDFILE method to add the template file to that collection." Here we have a Corvid Collection variable named "report" and we are adding the FO template named “ReportTemplate.fo"


Commands
Commands
Help
Variables
Blocks
Reset
External
Control
Results
Title
Reports
Email
Assign a Value to a Variable:
Derive the Value of a Variable Using Backward Chaining:
Variable:
All Confidence variables
All variables with Final Result flag set
All Collection variables
Variables fitting the mask:
Ask the User for the Value of a Variable:
Variable:
Cancel
OK

● Assign a Value to a Variable:

Variable: report

Method (Collection Only) ADDFILE

Expression: ReportTemplate.fo

Command: SET [report.ADDFILE] ReportTemplate.fo

Doing this causes a variety of things to happen automatically. As Corvid adds the template, it will look for all the [[..]] expressions. If Corvid has asked or derived the values for the variables, the [[..]] expression will be replaced by the value. If the value is not yet known, Corvid will use backward chaining to derive the values for the variables. This may involve asking questions, using various rules, etc. Once all the [[..]] variables in the template have been filled in, the report with all the values will be in the collection variable. The individual [[..]] expression values can be short or long. They can be a word or number or a long paragraph. FO XML tags can be included in the replaced value text, but this requires knowledge of FO syntax. You may need to include many [[..]] expressions to build a complex report with a lot of formatting.

A note on good system design, while it is certainly possible to just use the ADDFILE command and embedded [[..]] variables to drive the logic of a session, this can result in a system that may not be easy for others to maintain in the future. Since the need to set the variables is due to the ADDFILE, rather than commands in the command block, it may not be obvious to others why the system is running. If the ADDFILE is used to drive the logic, be sure to make some note that this is driving the asking and derivation of the variables.

The other approach to building the report in the Collection variable is to have various sections of logic add individual sections of FO code into the report. This approach allows more complex and dynamic reports, but requires detailed knowledge of FO XML syntax to produce an overall report that is syntactically correct. It is easiest to build the RTF sections in MS Word, convert them and then cut out the smaller sections with Notepad and include them individually. This can be complicated, and the single template method works quite well for most situations.

4. Save the Report

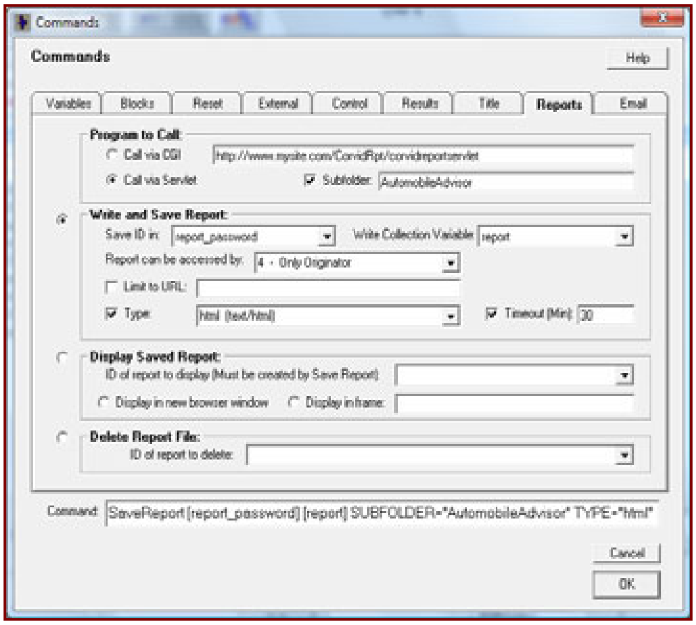
Once the report is in the Collection variable, it needs to be saved to the server and then displayed in a browser window using a special URL that will call a server program to display the report. Java security prohibits directly putting the report contents into a browser window, so the server route is required.

First build a SaveReport command. Click on the Reports tab in the Command Builder window.

- If you are using the Corvid Applet Runtime, enter the URL for the Corvid Report Servlet. This will be something like: http://www.myserver.com/CorvidRpt/corvidreportservlet.

- If you are using the Corvid Servlet Runtime, enter the URL for ExsysFop. This will be something like: http://www.myserver.com/ExsysFop/exsysfop.

Optionally, you can next enter a subfolder name. Subfolders provide a way to organize the report files that are created. This can be useful when there are multiple Corvid systems that build reports. A subfolder name is not required. The main location for stored reports is specified by the CorvidReport.cfg file. The 'Subfolder' field is optional and specifies a subfolder in the folder specified in CorvidReport.cfg. You may wish to use it if you have many expert systems and you want to store their reports separately. Create any referenced subfolders on the server – Corvid expects the subfolder to have already been created and will report an error if they are not. The subfolder is optional and there is nothing wrong with storing all reports in one folder.



Program to Call:

○ Call via CGI http://www.mysite.com/CorvidRpt/corvidreportservlet

● Call via Servlet ☑ Subfolder: AutomobileAdvisor

● Write and Save Report:

Save ID in: report_password Write Collection Variable: report

Report can be accessed by: 4 - Only Originator

☐ Limit to URL:

☑ Type: html (text/html) ☑ Timeout (Min): 30

○ Display Saved Report:

ID of report to display (Must be created by Save Report)

○ Display in new browser window ○ Display in frame:

○ Delete Report File:

ID of report to delete:

Command: SaveReport [report_password] [report] SUBFOLDER="AutomobileAdvisor" TYPE="html"

In the 'Write and Save Report' grouping, select the Collection variable that holds the contents of the report. You will need a String Variable to hold the ID for where the report is stored on the server. This can have any name, but should only be used to store the report ID. Select the String variable in the "Save ID in" dropdown list.

In the "Type" dropdown list, select the Type of report. For PDF reports, select "PDF" even though the report contents so far is FO. (For other types of reports, select the appropriate type).

If you wish to limit access to the URL that Corvid will create, you can limit the access group, but that is optional. (If you expect end users to send the report URL to others, this should NOT be selected.)

Click on the OK button.

5. Display the Report

The final step is to display the report.

For the Corvid Applet Runtime, change the program to call the ExsysFop program on the server. This will be something like: http://www.mysite.com/ExsysFop/exsysfop (Note: for all non-PDF reports, you would call the same CorvidRpt/corvidreportservlet as used in the SaveReport. Only PDF files require calling the special ExsysFop to do the FO to PDF conversion.)

Pull down the list for 'ID of report to display' and select the same String variable as used in the SaveReport command. The other options are optional. Click on the OK button.

For Corvid Servlet Runtime, it is a little more complicated. Unlike HTML reports, even though the URL to display the report is set by SaveReport, it cannot simply be used in DISPLAY_HTML because the content type will be incorrect. Instead the link has to be embedded in an HTML page that can link to the report.

For txt, rtf, xml, and pdf reports, the variable selected for ‘Save ID in’ contains a URL that will display the report. Create a web page that has a link that opens the report or uses Javascript to pop-up the report. The web page with the link could be the Results page.

```
SET [report.ADDFILE] ReportTemplate.fo
SaveReport [report_url] [report] TYPE="pdf" IP=4 SERVLET="http://host/ExsysFop/exsysfop"
RESULTS Servlet=ResultTemplate.pdf
```

The [report_url] variable in the example above will be assigned the URL that displays the PDF report. The URL calls CorvidRpt. The web page will have a link to the URL, like this:

```
<a href="[[report_url]]">Display Report</a>
```

The embedded link will actually be a call to ExsysFop, which will display the PDF file. When the user clicks on the link, whatever program their browser has associated with "pdf" documents (usually Adobe Reader) will display the PDF report.

To display the report in a pop-up window, change the <body> tag to this:

```
<body onload='window.open("[[report_url]] ","_new")'>
```

Delete the report

In most cases, the report should be left on the server so that the end user can bookmark the link for later reference. However, you can have the system delete the report after the user is done. This can be done with a DeleteReport command.

To build the DeleteReport command, make sure the URL in "Program to Call" is CorvidRpt and the subfolder is the same as for the SaveReport command. Pull down the list for ‘ID of report to display’ and select the same variable as in the SaveReport command. Click on the OK button. The DeleteReport command should not immediately follow the DisplayReport command, and should only be used after asking the user if they are done with the report. Most systems should not use DeleteReport and should instead delete old reports by other means such as periodically sorting the report folder by date and deleting reports that are too old.

Troubleshooting

Testing CorvidRpt

To test CorvidRpt directly on the server without using an expert system, open a browser and go to the url:

http://www.mysite.com/CorvidRpt/corvidreportservlet

A form will be displayed. Enter an ID, any word or random digits, for the value of Code1 and click the Submit button. This should create a very simple html report file on the server and send back a password that starts with the value of Code1 and some random digits and an ".html" extension. Copy this password and click on the browser's Back button. Paste the password in the Code2 field in the Display Report section. Type the same value of Code1 as before and click on the Submit button. It should display the report. For more elaborate tests, also paste the full text of the report into the "report" field and enter the extension of the type of document in "ext", for example, "rtf".

Can't Put .cfg File in WEB-INF

CorvidRpt and ExsysFop must know where the report is located. We put a .cfg file into the WEB-INF folder. If the ISP does not allow this or if using a servlet container, such as IBM WebSphere, which does not explode (unzip) the war file, then the .cfg file must be put inside the .war file. Here are the instructions.

1. Create a new empty folder, such as C:\scratch.
2. Copy the 'CorvidRpt.war' or 'ExsysFop.war' file to that folder.
3. Open a Command Prompt window in that folder (Start - All Programs - Accessories - Command Prompt)
4. Type cd and the path to the new empty folder. For example:

   cd \scratch
5. Type one of these:

   jar xvf CorvidRpt.war
   jar xvf ExsysFop.war

   If jar was not found, specify its path. Jar is included in the Java JDK, which is free from http://java.sun.com/.
6. Put the 'CorvidReport.cfg' file in the WEB-INF folder that was created.
7. Type one of these:

   jar uvf CorvidRpt.war WEB-INF\CorvidReport.cfg
   jar uvf ExsysFop.war WEB-INF\CorvidReport.cfg
8. Type: exit to close the Command Prompt window.
9. The war file has been modified. Deploy the modified war file to the server.

Error message: Undefined Embedded Variable

This occurs when the WYSIWYG program inserts formatting commands between the [[ ]] of an embedded variable in the template document. Word does this more than WordPad. Open the RTF or FO template in Notepad and search for "[". Check to see that embedded variables are of the form "[[varname]]" without any extra formatting commands. If there are problems, open the template in Word or WordPad, make sure the entire [[..]] expression is selected and apply a format. It is generally best to make the changes in the RTF template and then rebuild the FO template.

Embedding a List

If the list of items of unknown length requires formatting for each item, you can include the formatting by including the BEFORE and AFTER options, like this: [[list]BEFORE:<li> AFTER:</li>]

Some WYSIWYG editors will have difficultly with this because, before embedding, it is incorrect syntax. Sometimes you can work around it with html editors by using &lt; and &gt; instead of "<"and ">".

Embedding Numeric Variables

An embedded numeric variable, by default, embeds as a floating point number. If it was intended to embed as an integer, then use the .FORMAT property to remove the decimal points. Like this:

[[num.FORMAT #######]]

If a constant number of digits after the decimal point is desired, use a FORMAT like this:

[[num.FORMAT #######.00]]

Other Problems

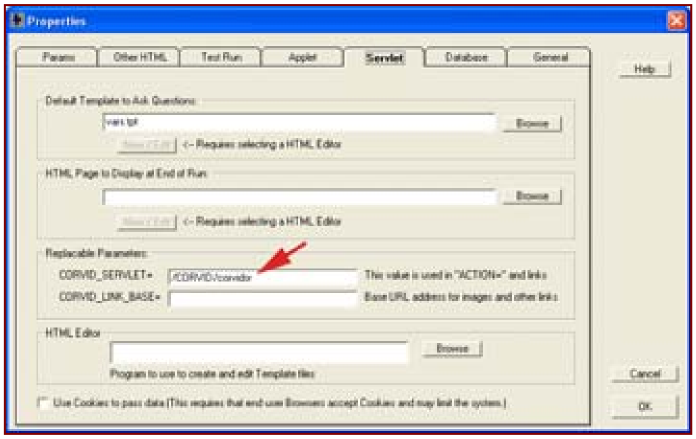
For other problems generating PDF reports, registered Corvid users should contact: **support@exsys.com**.

Changes to the CORVID_SERVLET Parameter

In versions of Exsys Corvid prior to v5.1, the CORVID_SERVLET parameter was used to indicate where the Corvid Servlet Runtime had been installed. The value was set from within the Parameters window.

The value set for CORVID_SERVLET was used in the template files for the "ACTION=" command to send the next set of data back to the Servlet Runtime. While this worked well, whenever a system was moved to a different server, the value for CORVID_SERVLET had to be edited using Corvid and the CVR file rebuilt.

In Corvid v5.1, the value for CORVID_SERVLET is taken from the initial URL used to start a Corvid system with the Corvid Servlet Runtime. Since this URL has the location of the servlet runtime, that is used as the value for CORVID_SERVLET.


Properties

Default Template to Ask Questions: vars.tpl

CORVID_SERVLET= /CORVID/corvidsr

CORVID_LINK_BASE=

□ Use Cookies to pass data (This requires that end user Browsers accept Cookies and may limit the system.)

Params
Other HTML
Test Run
Applet
Servlet
Database
General
Help
Browse
New / Edit
<- Requires selecting a HTML Editor
HTML Page to Display at End of Run
Browse
New / Edit
<- Requires selecting a HTML Editor
Replacable Parameters
This value is used in "ACTION=" and links
Base URL address for images and other links
HTML Editor
Browse
Program to use to create and edit Template files
Cancel
OK

The CORVID_SERVLET replaceable parameter is still used in the templates just as in previous versions, but it is not necessary to set the value in the Parameters window, and any value set there will be ignored. The correct value will always automatically be set.

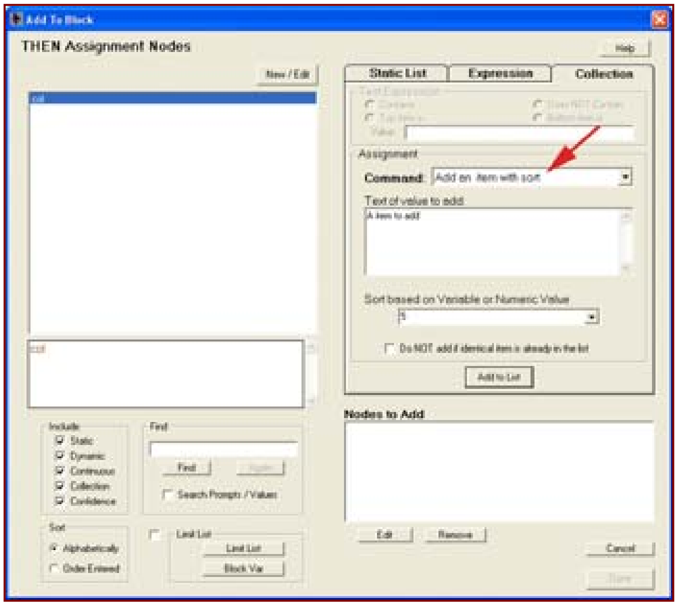
New .SCORE Property for Collection Variables

When the values for a Collection variable are added with “sort”, each item in the value list will have an associated sort score.

The order of the values in the list will be based on the sort scores, with the highest at the top of the list. This is generally the basis for any product selection system. While it was easy to get the top item(s) off the list, it was not previously possible to get their sort score. A new method:

[Collection.SCORE #]

will give you the score of the item number # in the value list. This is normally used to check the value for the top item to see how high a score it received. For example, if [recommendations.SCORE 1] is a high value, the system was able to find a good match to the user’s requests. If it is a low value, none of the items was a very good match, though the best one was still selected.


Add To Block

THEN Assignment Nodes

Command: Add an item with sort

Text of value to add: A item to add

Sort based on Variable or Numeric Value: 5

□ Do NOT add if identical item is already in the list

Include: ☑ Static ☑ Dynamic ☑ Continuous ☑ Collection ☑ Confidence

Sort: ● Alphabetically ○ Order Entered

The .SCORE property can be used anywhere a numeric value is allowed, such as:

IF

([recommendations.SCORE 1] > 100)

THEN …

Security Enhancement for CorvidDB

The version of CorvidDB included with Corvid v5.1 (CorvidDB.war) enhances security and prevents a possible security problem with earlier versions. If your Corvid systems make use of CorvidDB to access a database from your expert system, it is highly recommended you use the new CorvidDB.war.

You only need to use CorvidDB if your system uses the Corvid Applet Runtime and it reads or writes data to a database with SQL commands. In that case, it is highly recommended that the old CorvidDB.war file be replaced with the new CorvidDB.war file. You do NOT need to modify your system. Once installed, the new CorvidDB will automatically be used. The new CorvidDB.war file is backwards compatible with older systems. After the new CorvidDB is set up, it will work with existing expert systems.

The new CorvidDB.war file requires the .cdb files to be placed in the WEB-INF folder instead of the root of the servlet.

Here are the instructions for Tomcat:

- Copy the .cdb files from the old CorvidDB.war file's servlet root folder. You will find them in Tomcat's 'webapps/CorvidDB' folder. Put them in a temporary folder.
- Either stop Tomcat and delete the CorvidDB folder and the CorvidDB.war file or use the Manager web page to delete the CorvidDB servlet.
- Put the new CorvidDB.war file in the 'webapps' folder.
- Start Tomcat or use the Manager web page to deploy the CorvidDB servlet.
- Copy the .cdb files from the temporary folder and put them in Tomcat's 'webapps/CorvidDB/WEB-INF' folder.

If the .cdb files cannot be put into the WEB-INF folder because the ISP does not allow it or if using a servlet container such as IBM WebSphere, which does not explode (unzip) the war file, then the .cdb files must be put inside the war file. Here are the instructions:

1. Create a new empty folder such as C:\scratch.
2. Copy the ‘CorvidDB.war’ file to that folder.
3. Open a Command prompt window in that folder (Start – All Programs – Accessories – Command Prompt).
4. Type **cd** and the path to the new empty folder, for example: **cd \scratch**
5. Type this: **jar xvf CorvidDB.war** If jar was not found, specify the path. Jar is included in the Java JDK, which is free from: **http://java.sun.com**.
6. Put the .cdb files in the WEB-INF folder that was created.
7. Type this: **jar uvf CorvidDB.war WEB-INF*.cdb**.
8. Type: **exit** to close the Command Prompt window.
9. The war file has been modified. Deploy the modified war file to the server.

If you have any questions on installing CorvidDB, contact Exsys Technical Support for assistance at: **support@exsys.com**.

Exsys, Inc.
6301 Indian School Rd. NE, Suite 700
Albuquerque, NM 87110 U.S.A.
Tel: (505) 888-9494, Fax: (505) 888-9509

info@exsys.com
www.exsys.com

For support questions contact: **support@exsys.com**