ORACLE

Oracle® Documaker Desktop

Interfacing to External Systems via Import and Export Routines

Version 12.3

Part number: E52927-01

March 2014

CONTENTS

Chapter 1

Interfacing to External Systems via Import and Export Routines

You can use Documaker Workstation's (PPS) data import and export features to interface to external systems. During an import, the system automatically fills specific variable fields with data from another application.

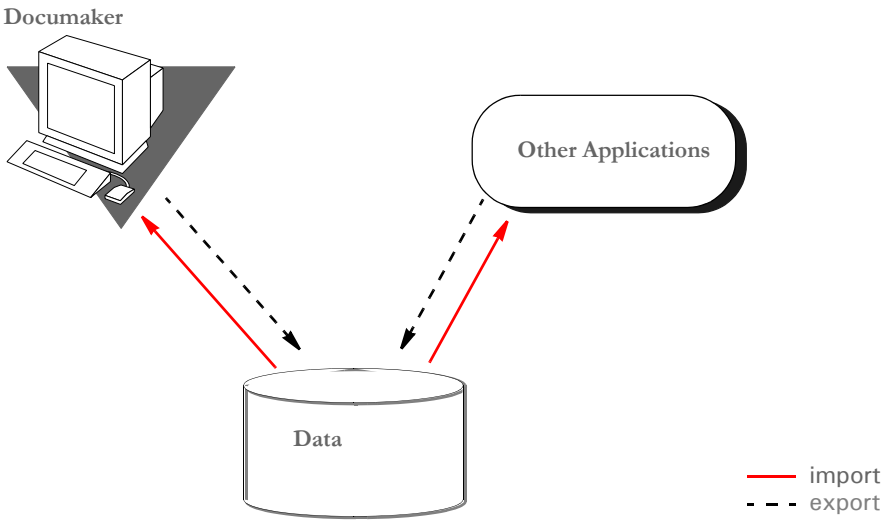
Likewise, you can export data from Documaker Workstation to other applications. The export process provides the same benefits as the import process.

This document includes information on…

Importing and Exporting Files

When users export data, they send a file to a specified directory so other applications can use that same data. The system's import and export functions work with the manual data entry features. You can import and apply common data, such as a name and address, then manually add the information specific to the form set. By importing common data and manually entering the remaining data, you can save time and reduce data entry errors.

The export function lets you extract data from the system, and then use the data in another application.



The system includes two import functions. You can use the Standard Import to import files which contain data for specific form sets. And, you can use a Selective Import to select import files that are specific to form set transaction types and user access levels.

NOTE: When you import multi-page images, the system includes after the field name a pipe symbol (|) with an *M* and the page number. Here is an example:

\NA=qvrfld|M2\

Standard import

A *standard import* file contains data for a specific form set. The import file contains a header record which the system uses to match the data with the appropriate form set. When you select a Standard Import function, the system displays a window which contains a list of import files. You must know what data the import files contain, and what type of form set transaction the data relates to, to choose the proper import file.

Selective import

A *selective import* function contains a driver file and import files. The driver file filters the import files based on the user's form set selections. The driver file recognizes the current user and the selected transaction type, and displays only the import files which correspond to that user ID and form set transaction. The driver file can include a brief description of each import file, which helps the user select the appropriate file.

Export file

An *export file* contains data from the system. You can apply a generic format to this data so other applications can read the exported data.

Setting Up the INI Files

You must perform these tasks to enable and configure the import and export functions.

- Modify the FSISYS.INI file
- Modify the FSIUSER.INI file

This chapter focuses on setting up your system so you can import and export information and outlines the settings your INI files must contain.

Each resource library contains two INI files which you must configure to use the import and export features. You enable or disable import and export options and select the import and export file formats in the FSISYS.INI file. In the FSIUSER.INI file, you specify the directory where the system stores the files.

NOTE: You must configure the INI files for (and in) each resource library for which you create import and export files.

Setting Up the FSISYS.INI File

When you set up your resource libraries, you may have several import files. If this is the case, your users can select from a list of files to import.

You enable or disable import and export functionality, and define the format of import and export files in the ImportFormats and ExportFormats control groups in the FSISYS.INI file. These control groups contain options which define import and export formats and function calls. The control groups contain option lines for specific import and export formats. Each option line includes these semicolon delimited parameters. Here is an example:

```
01=;TD;Standard Import;TRNW32->TRNImport;
02=;TD;Selective Import;TRNW32->TRNSelImport;
```

Syntax

```
;_NOT_USED;_DESCRIPTION;_DLL NAME->FUNCTION;
```

Parameter	Description
01	Any two-character numeric value for sorting purposes
NOT USED	This field is not used by the system. You can use this field for a code for the file format. If not used, place a semicolon to mark the field.
DESCRIPTION	The import format description which appears in the import window.
DLL-> FUNCTION	The Dynamic Link Library (DLL) and function to call when a user selects an import file. *Do not modify this parameter.*

If either the ImportFormats or the ExportFormats control group is missing from the FSISYS.INI file, or if a group contains no options, the import or export feature is disabled.

Configuring Import Format Options

Configuring import format options lets you enable and modify specific import formats. The SAMPCO resource library installed with the system contains a default import file format. You can modify the default formats to customize the import format selections displayed for the users. You can also modify the numeric value to display a format in a different order.

Setting Up Import Formats

Follow these steps to set up an import format:

1. Open the FSISYS.INI file in the resource library for which you want to use import files. You can use any text editor to open this file.
2. Locate the ImportFormats control group. Most text editors have a find or search function you can use to quickly find this group heading.
3. Add or delete the following lines, depending on the import format you want to use:

For this import format	Enter...
Standard	**01=;TD;Standard Import;TRNW32->TRNImport;**
Selective	**02=;SI;Selective Import;TRNW32->TRNSelImport;**

Modifying Import Format Settings

Follow these steps to change import format settings:

1. Open the FSISYS.INI file in the resource library for which you want to use import files. You can use any text editor to open this file.
2. Locate the ImportFormats control group. Most text editors have a find or search function you can use to quickly find this group heading.
3. Select the format option you want to modify.
4. Change only the numeric and the description parameters as desired, using the table in the previous topic as a guideline. The format description appears in the Import window when you select the import option.

Configuring Export Format Options

Configuring export format options lets you enable and modify specific export formats. You can create export files that are compatible with the import feature or using an older method with or without headers.

The standard export format is compatible with the import format. This format contains all of the key information used to create the form set, as well as, field data grouped by the form and image on which they occurred.

The older export methods are supported for legacy systems. If headers are included, the headers indicate the specific form set where the exported data originated. This format does not include any form information and omits much of the form set definition information. If you enable export files with or without headers, you should indicate this fact in the format description.

Setting Up Export Formats

Follow these steps to set up export formats:

1. Open the FSISYS.INI file in the resource library for which you want to use export files. You can use any text editor to open this file.
2. Locate the ExportFormats control group. Most text editors have a find or search function you can use to quickly find this group heading. Then add or delete the following lines to indicate the export formats you want to use:

For this export format	Enter...
Standard export format	**01=;V2;Export Version 2;TRNW32->TRNExportV2;**
Old method, without headers	**02=;NH;Export no Header;TRNW32->TRNExportNoNA;**
Old method, with headers	**03=;TD;Standard Export;TRNW32->TRNExport;**

Modifying Export Format Settings

Follow these steps to change export format settings:

1. Open the FSISYS.INI file in the resource library for which you want to use export files. You can use any text editor to open this file.
2. Locate the ExportFormats control group. Most text editors have a find or search function you can use to quickly find this group heading.
3. Select the format option you want to modify.
4. Change only the numeric value and the description, using the following table as a guide. The format description appears in the Export window when you select the export option.

Parameter	Description
01	Any two-character numeric value for sorting purposes
DESCRIPTION	The description of the import format which appears in the import window.

SETTING UP THE FSIUSER.INI FILE

Since both standard and selective imports use the same import files, you can store these files in the same directory. You specify the directory paths in the FSIUSER.INI file.

Configuring Import Options

There are several options you can use to configure the import feature so it works the way you need it to work. Simply omit any option you do not want to use. Here's how to find the section of the INI file which contains import options:

1. Open the FSIUSER.INI file in the resource library for which you want to use import files. You can use any text editor to open this file.
2. Locate the ImpFile_cd control group. Most text editors have a find or search function you can use to quickly find this group heading.

If not found, you will need to add this group to your INI file. Simply type in the text, exactly as shown above, and be sure to include brackets (< ImpFile_cd >).

The import options follow the group heading. Usually, the order in which the options appear does not matter. In some cases, however, INI options are dependent on other INI options. These dependencies are discussed in the description of each option.

Setting Up the Default Import File Name

```
< ImpFile_cd >
    File = file name
```

Substitute a valid file name of up to eight characters with a three-character extension where indicated. There is no default for this option.

The system uses the file name you enter as the default in the File window the user sees when selecting the import file.

Setting Up the Default Import Directory

```
< ImpFile_cd >
    Path = MSTRRES\(resource library name)
```

Substitute a valid drive and directory path where indicated. There is no default for this option.

Specifying the import directory lets you map where your import files are located. If you omit this option, the system looks for your import files in the working directory specified during installation.

The system uses the path you enter here to determine the default directory displayed in the File window. If you use the File option and the file name specified there also includes a path, that path *overrides* one entered here.

Setting Up the Default Import File Extension

```
< ImpFile_cd >
    Ext = .DAT
```

Substitute a valid file extension where indicated. Be sure to include the period. The default is *.DAT*.

On the File window, the system will use this default extension to fill the selection list with available files.

If you use the File option and the file name you entered there includes an extension, that extension overrides one entered here.

Setting Up the Driver File for Selective Import

```
< ImpFile_cd >
    SelectionFile = driver file name
```

Substitute a file name where indicated. This file name indicates the driver file you want to use for selective imports. There is no default for this option.

If you enter a valid driver file name here, the system will not display the File window. If you omit this option or if the file name you enter is invalid, the system will display the File window so the user can select a driver file to import.

Preventing Users from Changing Imported Data

```
< ImpFile_cd >
    ProtectFlds = Yes/No
```

Substitute Yes or No where indicated. The default is Yes, which means the contents of imported fields cannot be changed from within the system.

This option defaults to Yes because import data typically comes from another application. If the data is incorrect, most users prefer to correct the data in the original application and re-import the data into the system.

If you are using the import file to provide default data, and you do not mind if users change the imported data, set this option to No.

NOTE: Keep in mind that normally the system only lets users change a global field the first time the field is accessed. Your system may function differently if it has been customized.

Ignoring Invalid Groups in Import Headers

```
< ImpFile_cd >
    IgnoreInvalidGroup = Yes/No
```

Substitute Yes or No where indicated. The default is No. All Key1 and Key2 group names must correspond to those specified in the FORM.DAT file.

If you turn this option on, which it *is not* recommended, the system attempts to import the data into the current form set. Since the system does not check the form group information, there is no way to make sure the data in the import file is valid for the current form set. Any information in the import file which does not correspond with fields in the form set simply disappears.

Importing and Exporting a Form Set that has Images added using a DAL Function

You can use the DAL functions AddImage and AddForm to add images or forms to a form set and then export the information. To use this capability, you must use the Standard Export Version 2 export option and add the IgnoreInvalidImage option.

For example, first make sure your ExportFormats control group looks like the one shown here:

```
< ExportFormats >
    01 = ;V2;New Std Export w Header  ;TRNW32->TRNExportV2;
    02 = ;TD;Old Std Export w Header  ;TRNW32->TRNExport;
    03 = ;NH;Old Std Export w/o Header;TRNW32->TRNExportNoNA;
```

Then, add the IgnoreInvalidImage option to tell the import function to include any images added to the form set by the AddImage or AddForm DAL functions.

```
< ImpFile_CD >
   IgnoreInvalidImages = Yes
```

Be sure to set this option to Yes to avoid an error message. The default is No.

Configuring Export Options

As with import, there are several options you can use to configure the export feature so it works the way you need it to work. Simply omit any option you do not want to use. Here's how to find the section of the INI file which contains export options:

1. Open the FSIUSER.INI file in the resource library for which you want to use export files. You can use any text editor to open this file.
2. Locate the ExpFile_cd control group. Most text editors have a find or search function you can use to quickly find this group heading.

If not found, you will need to add this control group to your INI file. Simply type in the text, exactly as shown above, and be sure to include braces ([ExpFile_cd]).

The export options follow the group heading. Usually, the order in which the options appear does not matter. In some cases, however, INI options are dependent on other INI options. These dependencies are discussed in the description of each option.

Setting Up the Default Export File Name

```
< ExpFile_cd >
    File = file name
```

Substitute a valid file name of up to eight characters with a three-character extension where indicated. There is no default for this option.

The system uses the file name you enter as the default in the File window the user sees when selecting the export file.

Setting Up the Default Export Directory

```
< ExpFile_cd >
    Path = MSTRRES\(resource library name)
```

Substitute a drive and directory path where indicated. There is no default.

Specifying the export directory lets you map where your export files are located. If you omit this option, the system looks for your export files in the working directory specified during system setup.

The system uses the path you enter here to determine the default directory displayed in the File window. If you use the File option and the file name specified there also includes a path, that path overrides one entered here.

Setting Up the Default Export File Extension

```
< ExpFile_cd >
    Ext = .OUT
```

Substitute a valid file extension where indicated. Be sure to include the period. The default is *.OUT*.

On the File window, the user can use the List of Files field to display only those files with the export file extension.

If you use the File option and the file name you entered there includes an extension, that extension overrides one entered here.

Appending to an Existing Export File

```
< ExpFile_cd >
    AppendedExport = Yes/No
```

Substitute Yes or No where indicated. The default is No.

If you set this option to Yes and the export file exists, the system does not display the Confirm Overwrite message to the user. The system simply appends the current form set data to the existing file.

If the file name you specified does not exist, the system creates a new file using that name.

Suppressing the Confirm Overwrite Message

```
< ExpFile_cd >
    Overwrite = Yes/No
```

Substitute Yes or No where indicated. The default is No.

Enter Yes to turn off the Confirm Overwrite message if a user chooses an existing file as the export file.

The system ignores this option if the AppendedExport option is set to Yes. Appending to an export file implicitly means the export file will not be overwritten.

Suppressing the Export File Selection Window

```
< ExpFile_cd >
    SuppressDlg = Yes/No
```

Substitute Yes or No where indicated. The default is No.

If the File option was not defined, or if it was defined improperly, the window appears regardless of what you enter here.

The File Selection window is also affected by the AppendedExport option and the Overwrite option. If you set either of these options to Yes, setting the SuppressDlg option to Yes tells the system not to display the File Selection window.

If you set those options to No, the File Selection window appears unless the export file does not exist. This prevents the user from accidentally overwriting a valid export file without confirmation.

Controlling the Default Export Button on the Complete Window

Please note that this option *does not* appear under the ExpFile_cd control group. Instead, it appears in the Complete control group, which is usually located in the FSISYS.INI file.

```
< Complete >
    ExportOnComplete = Yes/No
```

By default, the check box is enabled and visible on the Complete window. The default is no (unchecked). If you set this option to Yes, the system checks the Export option on the Complete window.

Optionally, you can hide or disable the control so the user cannot change the setting. Hide the option by including *,hidden* after the answer, as shown below:

```
ExportOnComplete = Yes,hidden
```

To let the user see the option, but not change it, include *,disabled* after the answer:

```
ExportOnComplete = Yes,disabled
```

Exporting Recipient Information

Use this option to determine whether recipient information is exported. The default is No.

```
< ExpFile_CD >
    AFEExportRecips = Yes/No
```

If you set this option to Yes, recipients are written for each form listed inside angle brackets. Recipients are separated by commas and each recipient's copy count is shown in parentheses. This option lets you import recipient information that has been changed in a form set. For example:

```
\NA=\;SAMPCO;LB1;DEC PAGE;<AGENT(1),COMPANY(2),INSURED(1)>
```

If an image on a form has a recipient with a different copy count from the form, that image's differing recipients are written. Here is an example:

```
\NA=qmdc3\<COMPANY(1)>
```

There is no option for importing recipient information from the import file, since the information is either in the file or not.

Listing the Export Functions You want to Use

The system lets you specify a list of export functions you want the system to execute. You can use this capability for a variety of purposes. For instance, you can create a backup copy of an exported file in a second directory or you can run multiple exports using different export methods.

To install this feature, add the following INI option:

```
< ExportFormats >
    99 =;ME;Multi-Export;TRNW32->TRNMultiExport;
```

If you have other export functions listed under this control group and you want to remove those functions, insert a semicolon as the first character on the line to *comment out* that line. You can also delete the line from the INI file.

Keep in mind:

- The INI file is sorted when loaded. This lets you control which option appears first, second, third, and so on. For instance, you can use numbers as the option indicator left of the equal sign (01 =, 02 =, and so on) to set the order of the options.
    - The semicolons identify and separate each portion of the definition.
    - *ME* is a token that represents this export method. You can change the text, but make sure it is unique.
- The text *Multi-Export* appears in the Export area on the Complete Form Set window. You can enter any description you want, but space is limited in the export list display area. Make sure the text you enter fits.
- The final portion of the definition

    ```
    TRNW32->TRNMultiExport
    ```

    must be copied exactly as shown. This identifies the DLL to load and the function that performs this feature. This information is case sensitive.

Calling Export Operations

This export feature does not export data. Instead, it lets you call multiple export operations during a single event. For instance, you can use this feature to call the standard V2 export multiple times or in combination with other export routines.

Use the MultiExportList control group to list the export operations you want to call:

```
< MultiExportList >
    01 = TRNW32->TRNExportV2
    02 = TRNW32->TRNExportV2;EXPORT2;EXPFILE_CD;
    03 = TRNW32->TRNExportDS
```

The options in this control group are sorted by the value to the left of the equal sign.

Each option must list the export DLL and function name you want to execute. The export functions you list here do not have to be defined in the ExportFormats control group shown earlier. The method of defining this information is:

```
DLL->FunctionName
```

Notice the example above actually names the export function, *TRNExportV2*, twice in the list. Depending upon how you have configured your INI options for each export function, this may or may not result in a separate export file.

If you want to temporarily change the export options used by the called function, you can include two optional parameters, separated by semicolons, after the export function name. You can see an example of this in this option line:

```
02 = TRNW32->TRNExportV2;EXPORT2;EXPFILE_CD;
```

Each parameter names a control group that contains the options you want to temporarily use.

The first parameter names a *source* control group that contains alternate INI options you want to apply to the *destination* control group in the second parameter. The destination control group is the control group normally associated with the export operation you want to call.

The above example tells the system to copy the INI options found in the Export2 control group to the ExpFile_CD control group. The standard V2 export function, TRNExportV2, uses the ExpFile_CD control group to define its options.

During the first export, the options remain as is. During the second export operation, the system first copies the alternate options from the source control group into the destination control group used by the export function.

This option switching process replaces any INI options which have the same name and adds missing INI options to the destination group. Once the export function completes, the original INI values from the destination group are restored. This makes sure your default INI options are always left intact once the export operations are completed.

NOTE: Any INI options that exist in the destination group and are not replaced by an option in the source group are left unchanged. This means that you only have to include in the source control group those options you want to alter for the subsequent export operation.

Keep in mind...

- The individual export functions listed under the MultiExportList control group do not know they are being called indirectly or in series. Each operates normally unless you override their INI options by substituting INI options as discussed previously. Normal operation can include displaying a window to ask for an export file name or for permission to overwrite an existing file.

  So, if you want your exports to work without user intervention, check the INI options for each export and use those that answer or suppress questions which have to be answered for the export to operate.

- Each export operation can generate error messages. The MultiExport feature calls all export functions in the list regardless of any errors which occur inside the subordinate export functions. If one of the export operations fails, this does not necessarily prevent the other operations from being called.

- If an export function named in an option in the MultiExportList control group cannot be located, an error message appears and that function is skipped.

- If you define a *source* control group without naming a *destination* control group, that will cause an error and the export function will be skipped.

- Do not include the multi-export function in the list declared in the MultiExportList control group.

- Do not name the MultiExportList control group as a destination control group on any of the listed export functions.

- If the source and destination control group are the same, the export is called without doing any substitutions.

- If you specify a destination control group without specifying a source control group, the export function is called and any substitutions are omitted.

Export Functions and INI Control Groups

This table lists the various export functions and the corresponding control groups which contain the options that apply to these functions.

Function	Control group	Description
TRNW32->TRNExportV2	ExpFile_CD	The standard V2 (version 2) export.
TRNW32->TRNExportDS	ImpExpCombined	The full-document export or combined-WIP export.
TRNW32->TRNExportFields	ExportFields	The field data export used to limit exporting to certain fields and WIP record information.
WXMW32->WXMExportXML	XML_Imp_Exp	The XML document export function. This export is included with Documaker licenses, but sold separately with other products, such as PPS.

Using the Field-only Export Option

The standard export feature outputs all form and field information contained in a form set. In some cases, you may want to extract only certain field information (like accounting data) from the form set without having to deal with all of the field and header information included in the standard export. You can use the Field-only export option to limit the exported output to the fields you choose.

By installing and using the Field-only export option, you can create an export file which contains only the information you want. After you install this export method, the Field-only export option appears on the Complete Formset window, along with the other export options.

This export method lets you specify which fields should be written to the export file. Each field is located by name, regardless of which image contains the information. Rather than output the same field and data numerous times, this export option only writes the first occurrence of any specified field in the form set.

In addition, if you later plan to import this information using another program, you can define an *alias* name which can be written to the export file. This means that you are not limited to using the field names chosen by the form designers.

The field-only export option uses many of the same INI options specified for a standard export method. Detail information about all options are included in this topic.

Installing the Field-only Export Option

To turn on this feature, first locate the ExportFormats control group. This control group defines the export methods available to the user on the Complete Formset window. It should look something like this:

```
< ExportFormats >
    01 = ;TD;Standard Export;TRNW32->TRNExportV2;
```

Then follow these steps:

1. To add this new export option, edit the file and include the following line under the ExportFormats control group:

   ```
   02 = ;FX;Field-only Export;TRNW32->TRNExportFields;
   ```

   There are several important things to note about this new line.

   - The INI file is sorted when loaded. Therefore, you can control which option appears first, second, third, and so on. This is accomplished in this example by using numbers as the option indicator -- 01 comes before 02. If you wish the new option to be first, you can simply change the numbers used for each option.
   - The semicolons are necessary to identify and separate each portion of the definition.
   - The first bit of information, FX, is a token that represents this export method.
   - The option text that appears in the Export area on the Complete Formset window occurs next. This text can be any text description. Please note, there is limited space in the export list to display the text description, so test to make sure the entire description can be seen on the window.

- The final portion of the definition must be copied exactly as shown. This identifies the DLL to load and the function that performs this specific feature. This information is case sensitive, so you should spell it exactly as defined here—otherwise the option will fail to load when chosen.

2 Next, one of the INI files will need to be changed. The FSIUSER.INI file typically contains *individual* user options, while the FSISYS.INI file contains the *universal* options shared by all users. The file you elect to change should be based upon whether you intend for all users or a select few to use the field-only export option.

NOTE: Control group options in the FSIUSER.INI file work with options in the FSISYS.INI file and in some cases, override those settings. This means that if you define the same control group in both INI files, they combine to form one control group when loaded.

3 In the INI file you chose, add the ExportFields control group. For the settings that can be specified under this control group, see Setting Up the Field-only Export Option on page 15.

SETTING UP THE FIELD-ONLY EXPORT OPTION

This export feature supports many of the same INI options that can be specified for the standard export option. The main difference is that the control group name for this feature is *ExportFields* instead of *ExpFile_CD,* which is used for the standard export feature.

Specifying a default file name

```
< ExportFields >
   File = file name
```

There is no default for this option. If you omit this option, the user must specify a file name for export. If you enter a valid file name for this option, the File window used for selecting the export file name defaults to this value. Optionally, you can suppress the File window if a valid file name is specified.

Specifying a default path

```
< ExportFields >
   Path = path name
```

There is no default for this option and without it, the current working directory is assumed. Setting the Path option tells the system to automatically write the export files to this directory, or the initial directory presented in the File window.

NOTE: If you enter a file name *and* full path for the File option, that path overrides your entry for this option.

Specifying a default file extension

```
< ExportFields >
   Ext = .exp
```

The default for this option is *.EXP*. To use another extension, simply specify it using this option.

NOTE: If you enter a file name *and* an extension for the File option, that extension overrides your entry for this option.

Appending to an existing export file

```
< ExportFields >
    AppendedExport = Yes
```

The default setting is No. If you turn this option on and an existing file is selected (or specified by an INI option), no overwrite confirmation message is shown to the user. The current form set data is appended to the existing file.

Suppressing the overwrite file confirmation message

```
< ExportFields >
    Overwrite = Yes
```

The default is No. Enabling this option prevents the overwrite confirmation message from appearing if the user chooses an existing file as the export file.

This option is ignored if you activate the AppendedExport option. Appending to an export file implicitly means it will not be overwritten.

Suppress the Export File window

```
< ExportFields >
    SuppressDlg = Yes
```

The default is No, which means the window will appear. If an INI option has not been defined, or defined improperly for the default export file name, the window appears despite the setting of this option.

The File Selection window is also affected by the AppendedExport and Overwrite options. If either options is enabled (Yes) the SuppressDlg option prevents the File Selection window from appearing. If neither of these options are enabled, the window is only suppressed if the specified export file does not exist. This prevents users from accidentally destroying a potentially valid export file without confirmation.

Identifying the fields to export

```
< ExportFields >
    FIELD = field name ; alias
    FIELD = field name2 ; alias2
    ...
```

For each field that should be exported from the form set you must include a *FIELD=* line in the INI file. You may export as many fields as necessary, but each line must begin with the *FIELD=* statement.

The *field name* reference in the example should be replaced with an actual variable field name contained on a image within the form set. To determine the field names, it may be necessary to load the image in the Image Editor and check the properties of the given fields.

The alias reference is optional and should be separated by a semicolon from the image field name, if used. This identifies a different name to write in the export file, rather than the name of the actual field from the image. Use the alias feature when you are exporting information which will be imported into another application that uses different names for its fields.

For instance, assume the line reads as follows:

```
FIELD = Acct#;Account Number
```

This line would locate the Acct# field in the form set. If found, the data would be written to the export file using the name *Account Number*.

Other important notes about these settings:

- The order of the fields in the export file matches the order they are defined within the INI control group.
- If an alias is not provided, the exported field name will be the same as the image field name.
- Including a semicolon after the image field name, but not providing any text for the alias suppresses the output of a field name in the export file. The backslash, (\), will still precede the data for the field, even when the field name is suppressed.
- Any field that is not located within the form set is omitted in the export file.
- The field data written to the export file appears just as it does in the form set. Fields without data are written to the export file with no data following the backslash.
- When specifying an alias, spaces before and after the semicolon divider are ignored. This means that *IMGFLD;EXPFLD* is the same as *IMGFLD ; EXPFLD*.
- Image field names are not case sensitive. *IMGFLD* is equivalent to *imgfld* and will locate the same field within the form set. However, the case of the field name or alias in the INI file is preserved in the output file.

Defining header and trailer information

```
< ExportFields >
    START = text
    END   = text
```

These settings are optional and independent of each other. You can specify a START without an END or vice versa. No default is provided for the START and END options.

Since this export method does not include header information from the referenced form set, it may be necessary to identify the starting and ending locations of export information. This is probably most important when you are using the AppendedExport option, which means that more than one form set's information will be written to the same file.

The text specified on these options can be any ASCII string. The text is written exactly as it is specified in the INI file. START is written before the first field's data from the form set is written. END is written after all field data has been written.

The START and END text is written to the export file whether any field data is written from the given form set.

Exporting to a single line

```
< ExportFields >
    SingleLine = Yes (or No)
```

This options defaults to No and is not enabled unless you change it in the INI file.

Normally, each field's name and data is written to a different text line within the export file. By enabling the SingleLine option, this will condense the exported field information into a single line of output. This option is provided for those environments where this method of export is more desirable.

By default, when the SingleLine option is in use, each field name and data set is separated by a semicolon. You can change this field separator using the Separator option.

```
< ExportFields >
    Separator = text
```

This additional option is only recognized when you specify the SingleLine option. The text assigned to the statement is used to separate each field name and data set written to the export file.

The text can be any ASCII string value you want to specify. For example...

*Separator = **!**

tells the system to write the text

!

between each field. Here's another example...

Separator = NEXT FIELD=

tells the system to write the text

NEXT FIELD=

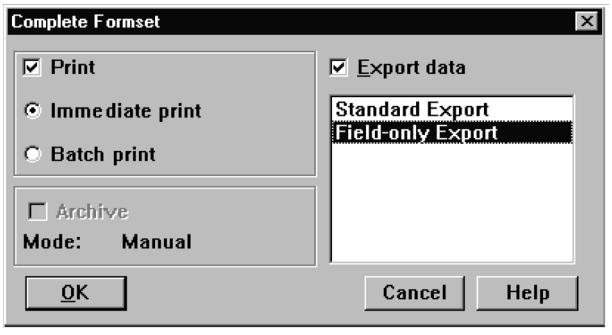
between each field.

Other important notes about these options:

- The separator text is only written when the SingleLine option is set to Yes.
- The separator text is only written between the field information sets. If you use a START option, the system omits the separator between the START text and the first field. Likewise, if you specify an END option, the system omits the separator between the last field output and the trailing text.
- The SingleLine option is probably of most value when using the AppendedExport option and writing more than one form set to the same export file.

Using the Field-only Export Option

Once installed, the option appears as a choice in the Export list, as shown below.



Choosing this option tells the system to write to the export file those fields specified in the ExportFields control group.

By default, the field information is written to the export file in the format shown below:

```
FIELD NAME\FIELD DATA
```

The field's name is separated from its associated data via the backslash (\) character. Using some of the other options provided with this feature, it is possible to change the output in several ways. For information about additional format options, see Setting Up the Field-only Export Option on page 15.

Setting Up Multiple Import Sessions

There may be occasions when users need to import data from more than one directory. For example, you may have one directory set up for form sets with a status code of *WIP* and another directory for form sets with a status code of *BatchPrint*.

To handle this requirement, you can define more than one set of options for auto-import in the INI file. The following examples show how to change the default AUTOIMPORT settings to accommodate this:

```
< TimerFuncs >
    01=;0;0;300;TRNW32->TRNAutoImport;
    02=;0;0;300;TRNW32->TRNAutoImport; \AUTOBATCH
```

In this example, definition *01=* assumes AutoImport is the control group that identifies the behavior for this auto-import registration. The second definition, *02*, names an alternate control group, called AutoBatch, which contains the definitions for that registration.

Each auto-import control group, such as AUTOIMPORT and AUTOBATCH in this example, can contain different auto-import options.

Notice that the same auto-import function is used for all definitions. You can set the TMRLIB flags (STATE, URGENCY, and SECONDS) as needed for each control group.

Once the system activates TRNAutoImport, it identifies the specific auto-import control group and options before it searches for import files.

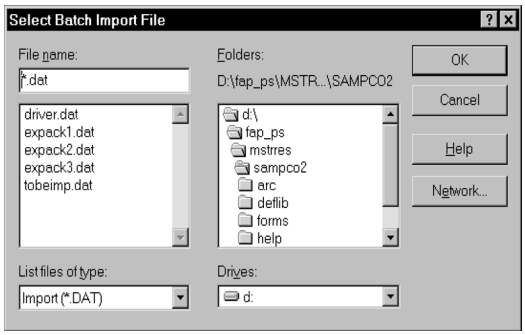
For more information about timed service functions, see the Documaker Workstation Supervisor Guide.

Batch Importing from a File

Batch importing from a file lets you import multiple form sets from a single file and place the individual form sets in the WIP list. When you batch import from a file, the system imports both the forms and the data. Users can then open the form sets from the WIP list and complete data entry. See Importing and Exporting Files on page 2 for more information on setting up import and export files.

Follow these steps to perform a batch import:

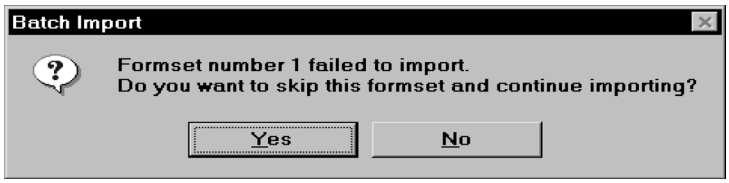
1. Choose Tools, Batch Import from File. The Select Batch Import File window appears.



2. Enter or select the batch import file. You can change the drive and directory if necessary. Click Ok.

NOTE: The system defaults to the working directory. You can change the drive and directory to locate a batch import file, but if you try to import form sets with key fields that do not match the current resource directory, the import will not be successful.

3. The system displays a message confirming the number of successful form set imports. Click Ok to close the message.
4. If there are data or key field discrepancies in any of the form sets, the system displays a warning message either telling you the form set number is not unique or stating there were no successful imports. If the problem is in a single form set, the system displays this message:



5. Click Yes to continue importing the form sets, or No to cancel the import. If you choose Yes, the system will place the form sets in the WIP list and display a message confirming the number of successful imports.
6. You can choose WIP, WIP List to view the form sets in the WIP list:

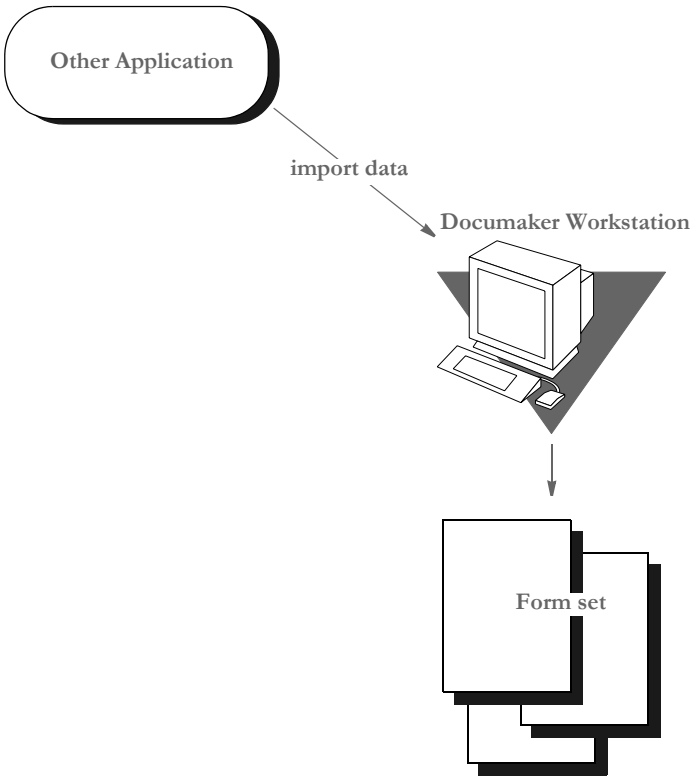
Creating a Standard Import File

Creating import and export files lets you automatically import data from an external application or data created during entry, into specific form sets in the system. You can also export data from the system to use in other applications or to import back into the system. When you create an import file, you must format the import data and set up the file so that the system knows which form set to apply the data to.

In addition to standard import files, you can also create a selective import driver file which displays applicable import files for user selection, based on the company, transaction, and user ID.

After you and the users have selected the forms for data import, your next task is to extract the applicable data from the host application system.

For a standard import file, you extract data to apply to a single form set. The type of data you import normally includes common data that does not vary. Common data can include company, line of business, policy, recipient, agent, lien holder, policy dates, and other general information. In addition, is it possible for the import file to designate which forms should be included in the form set. When importing data from a standard import file, the user must know what the import file contains to import it to the appropriate form set.



After you extract the host data, you must format it so the system can recognize and import the data to the correct forms, and to the correct fields on the forms.

You can format the text using any text editor. You may find it helpful to first map the data on paper, using an actual form as a guide, then enter the formats into the text editor.

NOTE: When creating files for batch import, follow the instructions for creating standard import and export files (with a header). For more information on configuring the system to append form set data to an existing export file, see Importing and Exporting Files on page 2.

For information on batch importing, see Batch Importing from a File on page 21. For information on exporting recipient information, see Exporting Recipient Information on page 11.

Sample Standard Import File

An import file is an ASCII text file which contains form set information. The file is formatted into lines with each line terminated by a carriage return/line feed (\r\n) combination. Each line is interpreted as one of these basic components:

- Header Record lines
- Form/image lines
- Field Definition lines

Some import files will only contain *header record* lines and f*ield definition* lines. Files may contain form/image lines if specific form information is to be included.

One header record is normally required, but any number of header records may be included. If more than one header record is included, they should all include the same KEY1 definition. Group all header records at the top of the file, because the system will not recognize additional headers once it encounters field or form information.

Header records consist of elements separated by semicolons (:). In addition, the line must start with a semicolon. The components of a header record are shown here:

```
;KEY1;KEY2;KEYID;TRANSCODE;STATUSCODE;DESCRIPTION
```

The first three components are required while the last three are optional. Note that all components are *positional*. This means that should a component be omitted, include the semicolon following the component. For instance, all of the following are valid header representations.

```
;KEY1;KEY2;KEYID;
;KEY1;KEY2;KEYID;TRANSCODE;
;KEY1;KEY2;KEYID;TRANSCODE;STATUSCODE;
;KEY1;KEY2;KEYID;;;DESCRIPTION
```

Component	Description
KEY1	A valid Key1 (Company) defined in the FORM.DAT file.
KEY2	A valid Key2 (Line of Business) defined in the FORM.DAT for that KEY1 (Company).
KEYID	the WIP KeyID (such as a policy number) provided on the Form Selection window.

Component	Description
TRANSCODE	Corresponds to a valid transaction type defined by the user in INI files.
STATUSCODE	A valid WIP status identifier which is also defined in the INI files.
DESCRIPTION	Text to be assigned as the WIP description.

Form/image lines define which forms and images on those forms to preselect for the user. Field information which is read after reading a form line is assumed to refer to fields on that form/image. Form lines have the following syntax:

```
\NA=name\;KEY1;KEY2;FORMNAME;
```

The first backslash (\) indicates this is a form/image line and must be followed by *NA*=.

KEY1 and KEY2 represent similar values to that shown for header records. Along with the FORMNAME (form's name), the system uses KEY1 and KEY2 to locate the specific FORM.DAT line that represents that form.

If name is included, it defines the image to locate on the current form. This component may be omitted to represent the form named for selection.

```
\NA=\;KEY1;KEY2;FORMNAME;
```

In addition, if the following line is included, this simply defines the next image on the previous form to locate.

```
\NA=name\
```

Field Definition lines refer to any line that does not begin with a semicolon or backslash. They take the following form:

```
FIELDNAME\FIELDDATA
```

Where FIELDNAME represents a field that is defined globally (if no current image or form has been named), or belongs to the last form/image that was named. The backslash (\ ) separates the field's name from the field's data.

FIELDDATA is any valid field (already formatted) for use by the named field.

```
POLICY CNT\31
POLICY EFFECT\04/25/96
POLICY EXPIR\04/25/97
```

In the sample data records above, each line is a field name, followed by field data:

- Policy Count = 31
- Policy Effective Date = 4/25/96
- Policy Expiration Date = 4/25/97

Field	Length	Description
Variable Field Name	32	Field name as it appears on the applicable form. Field names cannot contain a backslash (\).
Variable Field Data	255	Data corresponding to the field name. Field data cannot contain a backslash (\). Make field data lengths equal to or less than field lengths on the form. If data is greater than the form field lengths, the system truncates the data.

NOTE: If multiple fields with the same name are defined, the data in the last field with that name will import to all the similarly named fields in your form/image.

This example shows a formatted import file:

Header record line 1

;GENERAL AGTS INS;COMML PACKAGE;CPP_94;NB;WIP;Sample;

Header record line 2

;GENERAL AGTS INS;GENERAL;GEN_94;
INSURED NAME\Bob Bradshaw
POLICY NBR\332-224521-NR

Form/image line 1

\NA=IMG1\;GENERAL AGTS INS;COMML PACKAGE;FORM1;

Field data lines

POLICY CNT\31
EFFECT DATE\04/25/96
EXPIR DATE\04/25/97

Form/image line 2

\NA=IMG2\;GENERAL AGTS INS;COMML PACKAGE;FORM2;
POLICY CAN DATE\9/11/93
POLICY END DATE\9/11/94
POLICY AUD DATE\9/11/96

Form/image line 3

\NA=IMG2B\

Form/image line 4

\NA=IMG2B\
LOB1\AUTOMOBILE
PREMIUM 1\1111.1
POLICY FEE\44.4

Form/image line 5

\NA=IMG3\;GENERAL AGTS INS;GENERAL;FORM3;AGENT NUM\77
AGENT NAME\JOE SMITH
AGENT ADDRS1\1234 CANTERBURY LANE

Form/image line 6

\NA=IMG3\;GENERAL AGTS INS;GENERAL;FORM3;
AGENT CITY/ST\ANYWHERE/USA
AGENT PHONE\5551234
LOSS PAYEE\THE BANK, INC.

In this example, the form set contains two KEY2 groups (COMML PACKAGE and GENERAL) for the same KEY1. This normally indicates a packaged policy situation.

Field data defined before any form/image lines represents globally defined fields. Any form in the form set which has one of these globally defined fields will be prefilled with the imported data.

FORM1 and FORM2, are selected from the COMML PACKAGE group. Two copies of FORM3 are selected from the GENERAL group.

FORM2 identifies three images, IMG2, IMG2B, and IMG2B. The last two images are named the same, but no data is imported for the first occurrence of this image.

Rules for import files

Here are the rules for the import files:

- If no current form/image line has been identified, assume the fields are *form set global.*
- If a form is named and an image is not, assume fields have a *form* scope and will propagate to any identically named fields on that form.
- The first form/image line found must have the additional KEY1, KEY2, and FORMNAME fields. The program takes this information and searches the FORM.DAT for the appropriate form. Until another form is named, the system assumes the data following this line belongs to the named form/image.
- All forms identified by the import file are automatically selected for the user. This causes the form to appear with a check mark on the Form Selection window.
- Forms can be identified without additional image or field data.

Listing the global fields

You can use the OutputFieldScope option to list the global scope fields near the beginning of the FAP file:

```
< ExpFile_CD >
    OutputFieldScope = Yes
```

Here is an example:

```
;FORMMAKER PACKAGE;COMMERCIAL PACKAGE;EXPSCOPE=YES;NB;W  ;;
;FORMMAKER PACKAGE;GENERAL LIABILITY;EXPSCOPE=YES;NB;W  ;;
RENEWAL NBR\1098212-12
POLICY NBR\1098212
INSURED NAME\Fred Scope
ADDR1\1120 Global Scope Drive
ADDR2\Suite 3020
CITY\Atlanta
EFFECT DATE\02/17/01
EXPIRE DATE\02/17/06
BUSINESS DESCRP\Car Insurance
COMM PROP PREM\      1000.00
\NA=\;FORMMAKER PACKAGE;COMMERCIAL PACKAGE;FIL 1010 04 92;
\NA=\;FORMMAKER PACKAGE;GENERAL LIABILITY;CG DEC;
FORM LINE1\FCG 0001 04 93, FCG 0010 11 92, FCG 2100 01 93, FCD 0000
04 93
\NA=CGDEC~1\
\NA=\;FORMMAKER PACKAGE;GENERAL LIABILITY;FCG 0001 04 93;
\NA=CLF00\
\NA=CLF00~2\
\NA=\;FORMMAKER PACKAGE;GENERAL LIABILITY;FCG 0010 11 92;
\NA=\;FORMMAKER PACKAGE;GENERAL LIABILITY;FCG 2100 01 93;
\NA=\;FORMMAKER PACKAGE;GENERAL LIABILITY;FCD 0000 04 93;
\NA=CL000\
```

Global fields

Keep in mind this option only applies to standard V2 imports and that the system only lists the fields once per form set.

Testing a Standard Import Function

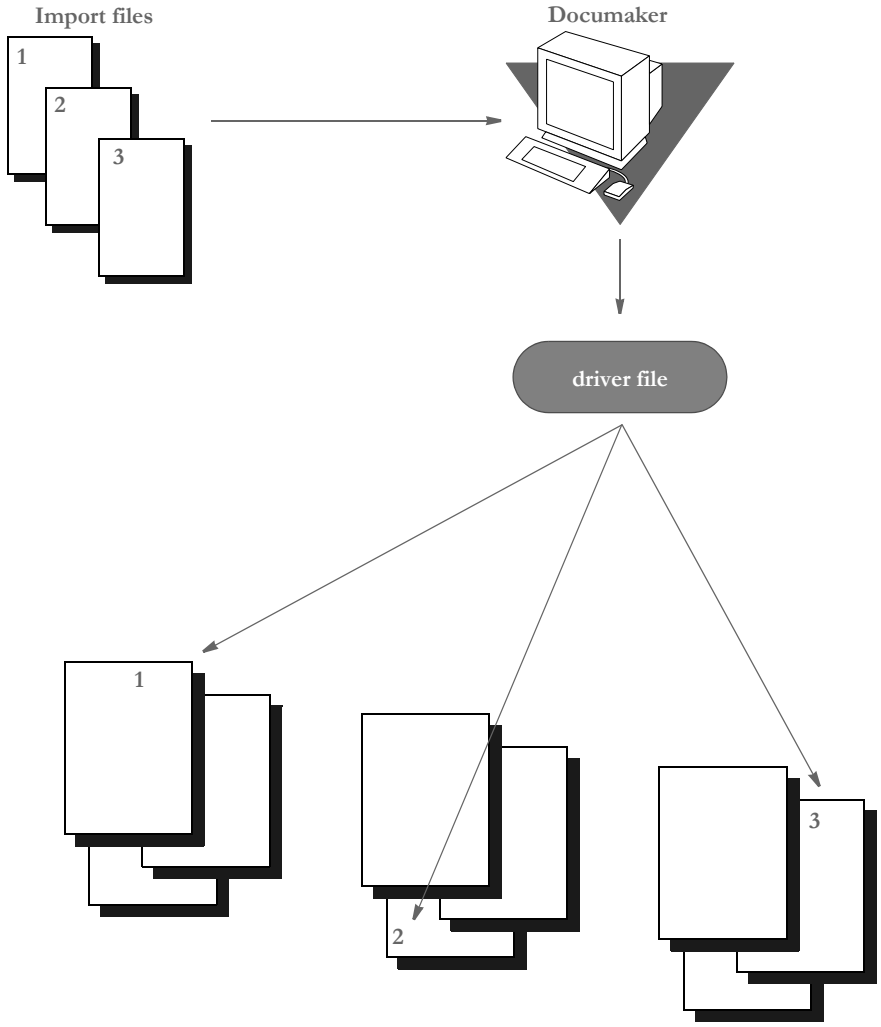
To make sure the data imports successfully, test the import files you create. Before you perform the test, first check the FSIUSER.INI and FSISYS.INI files to make sure import (and export) functions are enabled in the system. Also make sure at least one user ID is loaded into the system for import function initiation.

To test a standard import function, follow the steps for selecting forms, in Chapter 4 of the Documaker Workstation User Guide. If you created multiple import files, make sure you test them all.

Creating a Selective Import File

A selective import function uses a driver file that filters import data. When performing a selective import function, the user selects the driver file from the list of import files. The driver file recognizes the selected transaction type, the company name, and the user's access level, and displays only the import files that apply to that combination of variables.

If your company uses numerous import files, the user can narrow the file choices, to display only one or two that are appropriate for the selected form set type. Choosing a selective import function takes the guesswork out of identifying the correct file to import. The selective import function supports multiple users who access the driver file simultaneously.



SAMPLE SELECTIVE DRIVER FILE

The file format for selective import data is exactly the same as for standard import data. The import driver file format (fields) differs slightly. Like the standard import file, a driver file contains ASCII characters with semicolon delimited fields, and carriage return/line feed (\r\n) record delimiters.

Each record line in the driver file normally contains five fields: Resource Library (Company) Name, Policy Transaction Name, Import File Name (including the path), User ID, and File Description. The company library name entered should match the name of the resource library where you store the file.

These five fields indicate the form set that receives the import data. The file description indicates the form set that receives the import data, to ensure importing into the correct form set. Each two-line record indicates the particular form set to which you import related data. You cannot import form sets to different resource libraries. Here is a sample driver file.

```
;SAMPCO;NEWFMSET;\PPSWIN\MSTRRES\SAMPCO\
IMPORT3.DAT;DOCUCORP;General Information

;SAMPCO;ENDORSE;\PPSWIN\MSTRRES\SAMPCO\
IMPORT4.DAT;USER1;Health endorsement
```

NOTE: In the actual driver file, each record appears on a single line.

Formatting the Driver Import File

The table below contains syntax and format guidelines for the import driver file. Refer to this table as you create the driver records. You can assign any name to the driver file, as long as the users know the file name. Create the driver file as you would create a standard import file, using a text editor.

Field	Length	Description
Company	8	Company library name of the DOS batch file (no extension). This batch file calls a specific resource library directory (ppswin95\dll).
Transaction	8	Transaction type, enter: NEWFMSET - New form set ENDORSE - Endorsement RENEWAL - Renewal
Import File	127	Import file name, include the path and subdirectories
User ID	64	User ID for individual who completes the transaction
Description	35	Description of the import file Driver file parameters, appears in the Import Selection window

TESTING A SELECTIVE IMPORT

The requirements for testing a selective import function are the same as for a standard import function. Confirm that the import function is enabled in the FSIUSER.INI file for each company library you designated in the import driver file.

To test a selective import function, follow the steps for selecting forms, in Chapter 3 of the Documaker Workstation User Guide.

When a user selects the driver file from the Import File window, the driver file checks for the selected transaction type and user ID. The driver file then checks the transaction types you defined in each driver record for a match with the user selected transaction. By process of elimination, the driver file then displays only the import files applicable to the user's selected transaction type, and to the user's ID level.

Importing Information Directly into Archive

The system lets you import data directly into archive using a timed service function that automatically archives all records waiting in the Manual Archive queue.

This function performs the manual steps of choosing the WIP, Manual Archive option, selecting all records on the window, and then clicking Ok.

Use an INI setting similar to the following to set up this feature:

```
< TimerFuncs >
    08 =;0;0;100;AFEW32->AFEAutoArchive;
```

Where the first three values (*;0;0;100;*) represent:

;*State*;*Urgency*;*Seconds/Time-of-day*;*Function*

Value	Description
State	Always set to zero (0)
Urgency	Zero (0) or 1, where zero means the call can be skipped if the user is engaged in an operation and 1 means to call the feature as soon as the user has finished the current operation.
Seconds or Time-of-day	Enter the number of seconds to call the function at a specific interval, such as **300** for every five minutes. Or, enter the specific time of the day at which you want the system to call this function. The system uses a 24-hour clock, so enter **13:00** to indicate 1 pm.
Function	*AFEW32->AFEAutoArchive* indicates the timed service function. Enter this value exactly as shown. Without this value the automatic archive feature will never work. An error message will be displayed if this value is incorrect.

If you use this feature with the Automatic Import (to import records with a Manual Archive status), you can have the system archive imported files without requiring user intervention.

For more information on other timed service functions, see the Documaker Workstation Supervisor Guide.

Importing Global Data from Archive

This feature lets you select a form set from the Archive window and, instead of retrieving the entire original form set as it is done with Archive Retrieval, only import the global data from that form set. This data (without the forms) is then available for new form sets.

Using this feature, you do not have to keep the forms stored in the archive form set to make sure the data propagates onto the newer forms.

You set up the Import from Archive feature using INI files. The description of the Retrieve from Archive option in the Import list is based on your entries in the INI options.

To install the Import from Archive feature, add these INI options:

```
< ImportFormats >
    02 =;IA;Import From Archive;TRNW32->TRNImpDatFromArchive;
```

You can change the text that appears in the Select Import window as necessary.

NOTE: This feature does not select forms for you; it simply imports the data designated as global. You use the Scope field on the Attributes tab of the field's Properties tab to designate a field as global.

Importing Data with Forms and Images

If the form and image names match, the system can transfer data to the Entry system when importing forms and images from archive.

For instance, if the archived form set consists of *FORM_A* and *FORM_B* and you currently have *FORM_A* and *FORM_B_v2* selected, the system transfers the form data from *FORM_A*, plus any global fields set up for the entire form set. Since *FORM_B* does not match *FORM_B_v2*, no form or image level information is transferred between the form sets. Similar name matching is required at the image levels.

To use this feature, you must include these INI options:

```
< ImportFromArchive >
    SelectForms    = Yes
    TransferRecord = Yes
```

The default for both options is No, to be compatible with the original release of the ImportFromArchive feature discussed in the previous section (Importing Global Data from Archive on page 32).

If you set the SelectForms option to Yes, the import operation reselects the forms based upon those found in the archive. Any prior form select made by the user is removed.

If you set the TransferRecord option to Yes, the archive index record information is transferred to the WIP record information and the system updates the Form Selection window to show any changed values. For instance, you would set this option to Yes if you want the Key1, Key2, KeyID, and Description fields from the archived record used on your current form selection.

NOTE: Multi-line text fields will retain paragraph markers but will not retain any text formatting.

Exporting When Manually Archiving

You can have the system export a form set when it manually archives the form set. Using INI options, you can make the system wait until archive is about to occur before it exports the form set. This makes sure the exported data accurately represents the archived document.

The system checks the ArchiveExport option to determine which export routine you want to use. It then loads the form set into memory. This is necessary for the export function, but not for archival.

NOTE: Using the ArchiveExport method does not affect the export feature on the Complete window. This is a separate export from the one started there.

The export function is called before the archival functions are called. This way, if an error occurs during the export, the archival process stops and returns the error. The form set is still in WIP, so you can correct the error.

To use this feature, include the ArchiveExport option in the Complete control group, as shown in this example:

```
< Complete >
    ArchiveExport = DLL->FunctionName;SourceINI;DestINI;
```

The ArchiveExport option identifies the DLL and name of the export function you want to use. The export function does not have to be defined in the ExportFormats control group. Define the export function name the same way you define other entry hooks — a DLL name followed by the name of the name of a function to dynamically call to do the export. Here is an example:

```
< Complete >
    ArchiveExport = TRNW32->TRNExportV2;Export2;ExpFile_CD;
```

To handle different export options used when called in this manner, you can include two optional parameters after the export function name. Separate the parameters with semicolons. Each parameter names an INI control group that contain various options.

The first or source control group (*Export2* in the example) specifies the INI control group which contains alternative INI options you want to apply to the second INI control group (*ExpFile_CD* in the example). The second or destination control group should be the control group normally associated with the export function you want to use.

In the example above, the system copies the INI settings found under the Export2 control group to the ExpFile_CD control group. The standard V2 export function then uses the ExpFile_CD control group to find its options.

For instance, you might define the following:

```
< Export2 >
    Overwrite   = Yes
    SuppressDlg= Yes
    File        = ~HEXTIME .EXP
< ExpFile_CD]
    Overwrite   = No
    File        = Output.EXP
    Path        = .\data\
```

NOTE: You only have to include in the source control group options you want to add or replace options in the destination control group.

Once the export function completes, the system restores the original INI options and values from the destination group. This makes sure your default INI options are left intact once the export operation has finished.

So, using the example above, after the substitution takes place you would have the following options defined for the standard export:

```
< ExpFile_CD >
    Overwrite   = Yes
    SuppressDlg = Yes
    File        = ~HEXTIME .EXP
    Path        = .\data\
```

By restoring the original INI options and values for the destination control group, the system lets manually called export functions work with a standard set of options, while allowing the ArchiveExport method to be more automated.

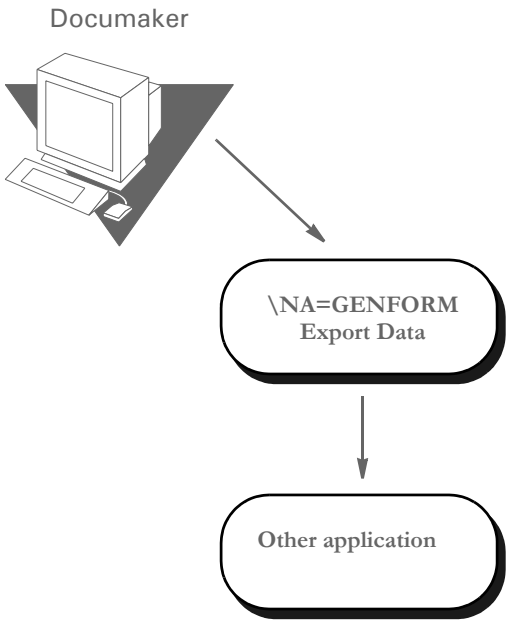
Keep in mind that the export function does not know it is being called indirectly. It will operate in its normal fashion. So, unless you override INI options as described here, the system may ask for an export file name or other information.

To further automate exports, check the INI options available for the export function and define those that answer or suppress questions for the export process so it can operate without user input.

Creating Export Files

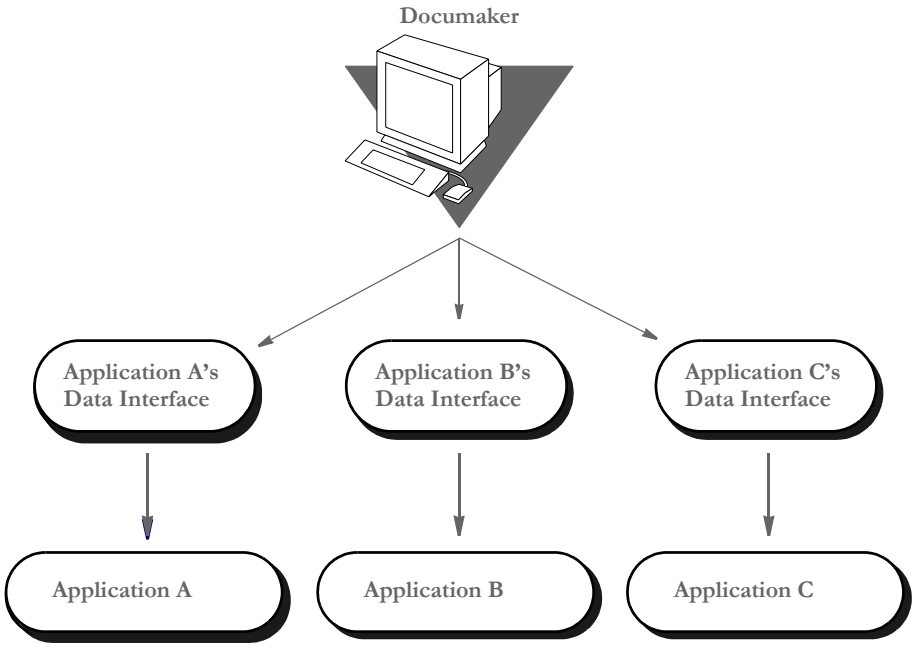
Creating an export file lets you extract form set data from the system and use the data in other applications or import the data back into the system. The export file contains the generically formatted system data that other applications read.

You can create two different kinds of export files: with headers or without headers. Depending on your system configuration, your export options may include Export Version 2. This export file is the same as the Standard Export (with a header). An export file with a header lets you indicate the form (or form set) where the data originated. An export file without a header does not include the form information. An export file without a header lets you export only the form data.



To create an export file, you can extract data that users have manually entered into the system, or data that was imported from another application.

The following illustration shows the flow of data when you create an export file without a header.



* The Application Data Interface is a program written by the application to read system export file data.

Creating an export file without a header lets you extract raw data from the system, without indicating the image where the data originated. With this option, users can use the exported data in various other applications.

Sample Export File Format (with Headers)

This example shows part of a sample export file with a header. The export file uses the exact format as a standard import file. Each record line contains the variable field name and the variable field data, separated by a backslash (\). The export file also contains the header as record line 1. If a form appears multiple times in a form set, the export file contains recurring variable field data for the duplicate forms.

Header record line 1

\NA=GENFORM\
INSURED NAME\John Smith
POLICY NBR\06871

Variable field data lines

PERSONAL LIMIT\300
EACH OCCR LIMIT\400
FIRE DAMG LIMIT\500
MED EXPEN LIMIT\600

Header record line 2

\NA=TESTFORM\
INSURED NAME\Richard Smith
POLICY NBR\28754

Variable field data lines

PERSONAL LIMIT\1000
EACH OCCR LIMIT\500
FIRE DAMG LIMIT\500
MED EXPEN LIMIT\2000

A form set may contain multiple forms. You can create an export file containing data corresponding to each form. The header records separate the data by image.

Below are two general guidelines to consider when creating export files, both with or without a header:

- If you create an export file corresponding to an import file, you can use the same name for both files, but use different extensions, or possibly use separate directories. For example…

```
(Library Name).IMP = Import Extension
(Library Name).OUT = Export Extension
```

- If you do not use an import file, the system prompts you to enter an export file name. If you use an import file, the system uses the file name of the import file also as the export file name.

Formatting the Export File

The following tables contain a section of the sample export file from the previous page, and the file format. Refer to these tables as you create the export data file. The headers separate the form data.

```
\NA=GENFORM\
INSURED NAME\John Smith
POLICY NBR\06871
PERSONAL LIMIT\300
\NA=TESTFORM\
INSURED NAME\Richard Smith
POLICY NBR\28754
PERSONAL LIMIT\1000
```

Field	Length	Description
Header	20	The name of the image in a form or form set where the export data originated. Header name is always in standard format: \NA= followed by the image name.
Variable Field Name	15	Field name as it appears on the applicable form. Same format as Standard Import file.
Variable Field Data	150	Data corresponding to the field name. Same format as Standard Import file.

Sample Export File Format (without Headers)

The format for an export file without a header is identical to an export file with a header, except the file does not include \NA records as headers. Each record line contains the variable field name, followed by the variable field data, separated by a backslash.

```
INSURED NAME\John Smith
POLICY NBR\06871
PERSONAL LIMIT\300
EACH OCCR LIMIT\400
FIRE DAMG LIMIT\500
MED EXP LIMIT\600
```

Field	Length	Description
Variable Field Name	15	Field name as it appears on the applicable form. Same format as export with Header file.
Variable Field Data	150	Data corresponding to the field name. Same format as export with Header file.

Testing an Export Function

Always perform a test export to make sure the export file you create contains the data you intended and the data is in the correct format. Test your application data interface to make sure your application system correctly imports the system data you exported. See the Documaker Workstation User Guide for instructions on importing and exporting.

Importing And Exporting WIP, NAFile, And POLFile Information

The Entry system includes an import/export format that combines WIP, NAFILE, and POLFILE information. This format exports information from the Entry system which will then be imported into Documaker.

This feature uses many of the same INI options as standard import and shares the same options with a standard export file (export with headers). The import function conforms to standard Documaker Workstation import requirements. The input to this function is a single file which contains WIP record information appended with standard NA and POL form set data. For more information on formatting import/export files, see Setting Up Import Formats on page 4 and Setting Up Export Formats on page 5.

To use this new format type, you must add the following options to your INI file:

```
< ImportFormats >
    02 = ;DS;Full Document Import;TRNW32->TRNImportDS;
< ExportFormats >
    04 = ;DSFull Document Export;TRNW32->TRNExportDS;
```

You can also add these options in the ImpExpCombined control group:

```
< ImpExpCombined >
    File       =
    Path       =
    Ext        =
    SuppressDlg=
    WIPHeader  =
    Separator  =
    Field      =
```

Option	Description
File	(Optional) Enter the name of the import or export file.
Path	(Optional) Enter the path for the import or export file if not included in the File option.
Ext	(Optional) Enter the default extension of the file. Defaults to *DS*.
SuppressDlg	(Optional) Set to Yes to prevent the File Selection window from appearing if enough information is specified in the INI file. The default is No.
WIPHeader	(Optional) Enter a text string to indicate where WIP record transactions begin. The default is *WIP*.
Separator	(Optional) Enter a text string to indicate an alternative separator for WIP fields. The default separator is a quotation mark (“).You could use another character as the separator, such as a semicolon (;).

Field	For each field that should be imported/exported to the WIP record line, you must include a line in the INI file. You can import as many fields as necessary. Begin each line with *FIELD*= and use this syntax:
	`Field=The WIP field name;stringformat`
	The string format is the standard C sprintf function format. Here are some examples:
	`Field=KEY1;%-3.3s`
	`Field=KEYID;%-10.10s`

For each field that should be exported to the WIP record line, you must include a line in the INI file. You may export as many fields as necessary. Begin each line with *FIELD=* and use this syntax:

```
FIELD=WIP field name;formatstring
```

Here are some examples:

```
FIELD=KEY1;%-3.3s
FIELD=;,
FIELD=KEYID;%-10.10s
```

Exporting Files Created by AutoImport, AutoPrint, or AutoArchive

When using AutoImport, AutoPrint, and AutoArchive, you can have the system automatically export the resulting file to another system. To use this feature, add the following option to your FSIUSER.INI or FSISYS.INI file:

```
< AutoImport >
    CompleteOnSuccess = Yes
```

NOTE: If you specify an alternate name for your AutoImport control group, add this new option under that group name.

The default for this option is No, which tells the system not to automatically export the file once the transactions have been processed.

Use these Complete control group options to automate the window in version 10.0.

```
< Complete >
    PrintOnComplete    = Yes
    ExportOnComplete   = Yes
    ArchiveOnComplete  = Yes
    SuppressDialog     = Yes
```

Set up the ExpFile_CD control group up to automatically complete.

```
< ExpFile_CD >
    File           = EXPORT
    EXT            = .OUT
    SuppressDlg    = Yes
    AppendedExport = Yes
```

NOTE: When operating in this manner, the first export function in your list is the one the system uses. Make sure you only have one export format defined, such as the ExportFormats control group, or define the default option so it appears at the top of the list.

Finally, go to your Printer groups and make sure these options are set:

```
< Printer >
    SuppressDialog = Yes
    PrtType        = PCL
```

NOTE: If you are using the GDI output or redirecting raw PCL through the GDI device, make sure you set the necessary options in that PrtType control group to automatically complete windows that appear.

Sending messages to a log file

Use the SuppressAllMessages option to send most error messages to a log file instead of having them displayed on screen. Keep in mind that unless you periodically check the log file for errors, you will not know errors are occurring. Here is an example:

```
< AutoImport >
    SuppressAllMessages = Yes
```

Working With XML Files

You can now import and export XML files while using Documaker Workstation and you can send and receive XML messages. Setting up the new import and export capabilities is similar to setting up any import/export file format.

NOTE: The ability to work with XML files is included in Documaker Workstation, but must be purchased separately by PPS users. You must also have a Docupresentment license to use the messaging features in the WXMEntryHookExtXMLLoad function because it calls Docupresentment files. Contact your sales representative for more information.

This feature uses these functions:

Function	Description
WXMImportXML	This function lets you import data from an XML file into a form set.
WXMExportXML	This function lets you export data from a form set to an XML file.
WXMEntryHookExtXMLLoad	This function lets you send messages from the system to any type of message server.

Setting Up Documaker Workstation

To use the import and export functions, you must also add this control group and options to your FSISYS.INI or FSIUSER.INI file:

```
< XML_IMP_EXP >
    Ext         =
    File        =
    Path        =
    SuppressDlg =
```

Option	Description
Ext	(Optional) Enter the extension for the output files. The default is XML.)
File	(Optional) Enter a file name, such as XMLEXP. If you omit this option the system prompts the user to enter the file name.
Path	(Optional) Enter the path, such as \xmlfile. If you omit this option, the system defaults to the current directory.
SuppressDlg	(Optional) Enter Yes to suppress the File Selection window. The default is No.

Follow the instructions below to complete the import, export, and messaging setup.

Setting up the XML export format

Follow these steps to set up the XML export format:

1. Open the FSISYS.INI file in the resource library for which you want to use export files. You can use any text editor to open this file.
2. Locate the ExportFormats control group. Most text editors have a find or search function you can use to quickly find this group heading. Then add the following line:

For this export format	Enter...
XML	**09=;XM;XML Export;WXMW32->WXMExportXML**

This assumes **09** is not already being used. Here is an example:

```
< ExportFormats >
    09=;XM;XML Export;WXMW32->WXMExportXML
```

Setting up the XML import format

Follow these steps to set up the XML import format:

1. Open the FSISYS.INI file in the resource library for which you want to use export files. You can use any text editor to open this file.
2. Locate the ImportFormats control group. Most text editors have a find or search function you can use to quickly find this group heading. Then add the following line:

For this import format	Enter...
XML	**09=;XM;XML Import;WXMW32->WXMImportXML**

This assumes **09** is not already being used. Here is an example:

```
< ImportFormats >
    09=;XM;XML Import;WXMW32->WXMImportXML
```

Setting up the XML message format

To send a message from Documaker Workstation to a message handling program such as IDS or MQSeries, you must add the EntryFormset INI option. Follow these steps:

1. Open the FSISYS.INI file in the resource library for which you want to use export files. You can use any text editor to open this file.
2. Locate the AFEProcedures control group. Most text editors have a find or search function you can use to quickly find this group heading. Then add the following option:

For this option	Enter...
EntryFormset	**WXMW32->WXMEntryHookExtXMLLoad**

Here is an example:

```
< AFEProcedures >
    EntryFormset = WXMW32->WXMEntryHookExtXMLLoad
```

CONFIGURING A 3RD PARTY INTEGRATION APPLICATION

Documaker Desktop can launch a 3rd party application to initiate the creation of a new document via the XML import function. You can use the 3rd party application to capture data and generate an XML import file. Documaker Desktop can then import the XML file as a new transaction.

Target 3rd party applications are those which capture or get policy data based on elements input into the application and which return a results code and an XML file in an import format Documaker Desktop can process.

Here are an examples of the INI options you use to configure a 3rd-party integration application:

```
< ImportFormats >
05 = ;XE;XML Import from EXE;wxmw32->WXMImportXMLFromExe
```

WXMImportXMLFromExe is the import function that launches 3rd-party options. You must also set up the following information about the 3rd-party application in the FSIUSER.INI or FSISYS.INI file. This control group is specific to the WXMImportXMLFromExe function.

```
< ImportXMLFromExe >
        Executable = \\(full path)\3rdParty.exe
        CurrentDirectory = \\(full path)
```

Executable option is used to configure the full path and file name for the 3rd-party executable and CurrentDirectory option allows to configure the working directory for the 3rd party executable. The 3rd parth application should output to the standard Output stream (stdout) the following information:

(return code) | (path to import file)

Insert the return code table contained in the feature 2807 document here and the example of the standard output file.

This <ImportFormats> option can also be used to allow the user to select a transaction from the standard (smart) archive and imports the data into a new transaction. This imports the data from an archived transaction and only optionally selects the forms if they are still defined in the current system. You may have newer forms that inherit the data. The format as follows

05 = ;IA;Import From Archive;TRNW32->TRNImpDatFromArchive;

Example

```
< ImportFormats >
   05 = ;XE;XML Import from EXE;wxmw32->WXMImportXMLFromExe
```

Setting Up IDS

If you are using IDS as the message server, you must also add the INI options shown below to let Documaker Workstation retrieve an archived record from IDS and load data into a form set before any data is entered by a user.

The archived record is retrieved using the Key1, Key2, and KeyID entered on the New Form Set window. For this to happen, you must set up the following request type in the DOCSERV.INI file for IDS:

```
< ReqType:GetXML>
    function = atcw32->ATCLogTransaction
    function = atcw32->ATCLoadAttachment
    function = atcw32->ATCUnloadAttachment
    function = dprw32->DPRSetConfig
    function = dprw32->DPRLocateOneRecord,Key1,Key2,KeyID
    function = dprw32->DPRRetrieveFormset
    function = dprw32->DPRPrint
    function = dprw32->DPRProcessTemplates
    function = atcw32->ATCSendFile, DOCC_XML, SENDBACKPAGE, TEXT
```

You can use any name for the archive library, as long as the same MRL name is used in Documaker Workstation.

You can set up the new function as an entry hook:

```
< AFEProcedures >
    EntryFormset = WXMOS2->WXMEntryHookExtXMLLoad
```

If you set it up as an entry hook, you must also set up these INI options:

```
< XML_IMP_EXP >
    DSIUseNTUserID        =
    DSIVARS               =
    DSIIgnoreTimeoutError =
    DSIAttachedVarFile    =
    DSIImportLevel        =
    DSITimeout            =
    DSIReqType            =
    DSIRecordDFD          =
```

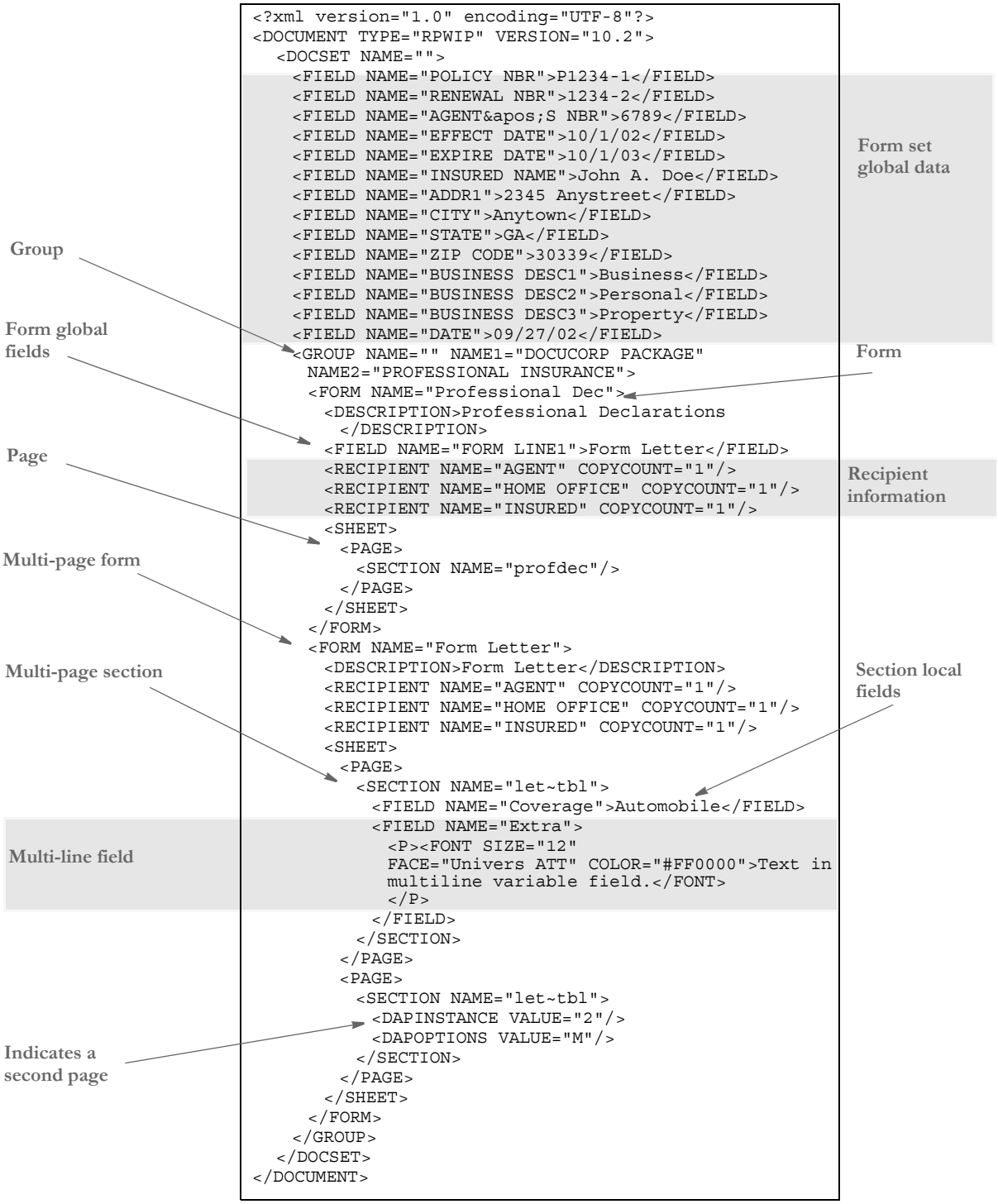
Option	Description
DSIUseNTUserID	(Optional) Set this option to Yes to use the NT user ID. The default is No. This gives you a way to pass the NT user ID in the queue instead of the normal DMWS ID.
DSIVARS	(Optional) Enter *variable*;*value*, where *variable* is the variable name and *value* is its value. This lets you identify a constant list of variables to be sent in the queue.
DSIIgnoreTimeoutError	(Optional) Enter Yes to continue processing if a timeout occurs. The default is No. This gives you a way to ignore a timeout when waiting on a return queue.
DSIAttachedVarFile	(Optional) The default is DOCC_XML. Set this option to the attachment name if it differs from DOCC_XML. This gives you a way to specify the variable name the XML file is attached to.
DSIImportLevel	(Optional) This option is typically used by programmers. Enter 2 if you want the hook to operate on the FAP_MSGOPEN level. Enter 3 if you want it to operate on the FAP_MSGRUN level. The default is 2.

Option	Description
DSITimeout	(Optional) Enter the number of milliseconds you want for the timeout. The default is 60000 milliseconds or 60 seconds.
DSIReqType	(Optional) Enter the name of the request type of the message placed in the queue. The default is GETXML.
DSIRecordDFD	(Optional) Enter the name of a DFD file. The system tries to match variable fields sent in the request to field values in this DFD file. It then attaches the DFD record to the end of the message.

If the request for an XML file comes back with an error, as opposed to a time-out, IDS displays an error message.

XML File Format

Here is an example of the format of the XML file the system creates:




```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT TYPE="RPWIP" VERSION="10.2">
  <DOCSET NAME="">
    <FIELD NAME="POLICY NBR">P1234-1</FIELD>
    <FIELD NAME="RENEWAL NBR">1234-2</FIELD>
    <FIELD NAME="AGENT&apos;S NBR">6789</FIELD>
    <FIELD NAME="EFFECT DATE">10/1/02</FIELD>
    <FIELD NAME="EXPIRE DATE">10/1/03</FIELD>
    <FIELD NAME="INSURED NAME">John A. Doe</FIELD>
    <FIELD NAME="ADDR1">2345 Anystreet</FIELD>
    <FIELD NAME="CITY">Anytown</FIELD>
    <FIELD NAME="STATE">GA</FIELD>
    <FIELD NAME="ZIP CODE">30339</FIELD>
    <FIELD NAME="BUSINESS DESC1">Business</FIELD>
    <FIELD NAME="BUSINESS DESC2">Personal</FIELD>
    <FIELD NAME="BUSINESS DESC3">Property</FIELD>
    <FIELD NAME="DATE">09/27/02</FIELD>
    <GROUP NAME="" NAME1="DOCUCORP PACKAGE"
     NAME2="PROFESSIONAL INSURANCE">
     <FORM NAME="Professional Dec">
      <DESCRIPTION>Professional Declarations
       </DESCRIPTION>
      <FIELD NAME="FORM LINE1">Form Letter</FIELD>
      <RECIPIENT NAME="AGENT" COPYCOUNT="1"/>
      <RECIPIENT NAME="HOME OFFICE" COPYCOUNT="1"/>
      <RECIPIENT NAME="INSURED" COPYCOUNT="1"/>
      <SHEET>
       <PAGE>
        <SECTION NAME="profdec"/>
       </PAGE>
      </SHEET>
     </FORM>
     <FORM NAME="Form Letter">
      <DESCRIPTION>Form Letter</DESCRIPTION>
      <RECIPIENT NAME="AGENT" COPYCOUNT="1"/>
      <RECIPIENT NAME="HOME OFFICE" COPYCOUNT="1"/>
      <RECIPIENT NAME="INSURED" COPYCOUNT="1"/>
      <SHEET>
       <PAGE>
        <SECTION NAME="let~tbl">
         <FIELD NAME="Coverage">Automobile</FIELD>
         <FIELD NAME="Extra">
          <P><FONT SIZE="12"
          FACE="Univers ATT" COLOR="#FF0000">Text in
          multiline variable field.</FONT>
          </P>
         </FIELD>
        </SECTION>
       </PAGE>
       <PAGE>
        <SECTION NAME="let~tbl">
         <DAPINSTANCE VALUE="2"/>
         <DAPOPTIONS VALUE="M"/>
        </SECTION>
       </PAGE>
      </SHEET>
     </FORM>
    </GROUP>
  </DOCSET>
</DOCUMENT>
```

Keep in mind...

- DAPOPTIONS should have a value of *M* for multi-page sections (FAP files). There are other section options, but only *M* is applicable in XML.

  Use DAPINSTANCE to provide a page number for multi-page sections. If the section does not span multiple pages, omit the DAPINSTANCE value.

- When you have multiple XML transactions within a single file, separate each transaction with a line feed. This is a requirement of Documaker software, not the XML parser.

- Although you do not have to include line feeds inside the XML for a transaction, we suggest you add a line feed after each element tag. This makes it easier to read the file and helps in debugging your XML. A message like

  ```
  Line 255, column 8, syntax is incorrect
  ```

  is easier to diagnose than

  ```
  Line 1, column 156780, syntax is incorrect.
  ```

Multiple User and Networking Issues

In a networked, multi-user environment you must carefully handle files to avoid conflicts. Two users trying to write to the same file at the same time is an obvious problem. In addition, problems can arise if one user tries to read a file, while another is writing to it. The import and export features of the system avoid these types of file conflicts.

When a user imports a file, the system opens the file and prevents or delays other users from writing to that file. Likewise, while a user is writing and exporting a file, other users are prevented or delayed from reading or writing to that same file. Multiple users can, however, read from the same import file at the same time.

When a conflict occurs, the workstation attempts to secure the file (in the proper mode) for up to 15 seconds. If the file becomes available within that time, the requested operation continues as expected. If, however, the file does not become available, the operation fails and the system displays a message. If the user is exporting a file, the message tells the user the export failed and asks if the user wants to try again. If the user is importing a file, the message simply states that the import file is invalid.

To change the delay or wait time, set the FSIWAIT environment variable to the number of seconds you want the system to wait. Place this setting in your AUTOEXEC.BAT file. For example, enter…

```
SET FSIWAIT=number of seconds
```

If you set FSIWAIT too low, operations may fail more frequently on a heavily used network. If you set it too high, the user may become concerned the program is no longer functioning.

NOTE: If you set FSIWAIT to zero, the system uses the default of 15 seconds instead. The least amount of wait time you can specify is one second.

Most of the time, users will be unaware of any delay because the import and export processes typically take only a second or two. The file is then released to other users.

Problems may still arise in environments where users open the import or export file in a file editor or some other program that keeps the file opened.