# Rocket LegaSuite 5250 Collector

## User's Guide

*Version 8.2*

September 2017
LCOL-8201–UG-01

# Contents

# Notices

## Edition

**Publication date**: September 2017
**Book number**: LCOL-8201–UG-01
**Product version**: Version 8.2

## Copyright



## Trademarks



## Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## License agreement



---

**Note:** This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when exporting this product.

---

# Corporate information

Rocket Software, Inc. develops enterprise infrastructure products in four key areas: storage, networks, and compliance; database servers and tools; business information and analytics; and application development, integration, and modernization.

Website: www.rocketsoftware.com

Rocket Global Headquarters
77 4th Avenue, Suite 100
Waltham, MA 02451-1468
USA

To contact Rocket Software by telephone for any reason, including obtaining pre-sales information and technical support, use one of the following telephone numbers.

| Country | Toll-free telephone number |
|---|---|
| United States | 1-855-577-4323 |
| Australia | 1-800-823-405 |
| Belgium | 0800-266-65 |
| Canada | 1-855-577-4323 |
| China | 400-120-9242 |
| France | 08-05-08-05-62 |
| Germany | 0800-180-0882 |
| Italy | 800-878-295 |
| Japan | 0800-170-5464 |
| Netherlands | 0-800-022-2961 |
| New Zealand | 0800-003210 |
| South Africa | 0-800-980-818 |
| United Kingdom | 0800-520-0439 |

## Contacting Technical Support

The Rocket Community is the primary method of obtaining support. If you have current support and maintenance agreements with Rocket Software, you can access the Rocket Community and report a problem, download an update, or read answers to FAQs. To log in to the Rocket Community or to request a Rocket Community account, go to www.rocketsoftware.com/support.

In addition to using the Rocket Community to obtain support, you can use one of the telephone numbers that are listed above or send an email to support@rocketsoftware.com.

# Chapter 1: About LegaSuite 5250 Collector

The **LegaSuite 5250 Collector** is a tool provided by Rocket Seagull for zSeries and IBM iSeries. You will use it to scan host application display files and menus. It will then collect the application screens and messages.

## Why Use the Collector

The Collector allows you to do the following:

- Specify the libraries you want to Collect,
- Set up user options,
- Run the Collector process in batch mode,
- Upon completion of processing, the result is stored in the `AWHOST` file.

## Installing the LegaSuite 5250 Collector

Consult the *Installation Guide* PDF in the **Documentation** folder for steps on installing the LegaSuite Collector for the iSeries.

## Before Using the LegaSuite 5250 Collector

After installation, it is wise to perform the following checks before using the LegaSuite 5250 Collector .

1. Check if there is an existing LegaSuite 5250 Collector,
2. Check if there are any other users currently using the existing LegaSuite 5250 Collector on the host.

### 1. Checking for Installed Collectors

Perform the following task(s) to check if there is a LegaSuite 5250 Collector already installed:

Type in the `WRKLIBPDM SEACOL` command which shows if one or move versions of the LegaSuite 5250 Collector are present on the host.

### 2. Checking if the Existing Collector is in Use

Perform the following task to check if any other LegaSuite 5250 Collector users.

If library **SEACOL** is present, use the `WRKOBJLCK OBJ(SEACOL) OBJTYPE(*LIB)` command to see if any user and/or job is using the LegaSuite 5250 Collector at this moment.

# Chapter 2: Tips for Using the Collector

## How to Use the Resulting AWHOST File

The AWHOST file must be transferred to a LegaSuite Builder where a Picture Album is created from the AWHOST file.

The Picture Album is evaluated for unique picture identification. If too many pictures are not uniquely identified, these pictures would require a manual identification process. When the amount of pictures which are unidentified is too large, you should influence the Collector with User Options Parameters and Accessing Collector Proposals, on page 22 to improve identification of your host screen formats to use (and omit or combine).

## Re-running the LegaSuite 5250 Collector

It is useful to know when to re-run the LegaSuite 5250 Collector.

Host application updates may also require re-running the Collection process. In this case, you can elect to rebuild the entire application or specify (by date) to collect on recently changed objects. In either case, you will need to transfer the AWHOST file to the Builder again.

**Note:** Before re-running the Collector, always save your AWHOST file for future reference.

# Chapter 3: About the Collector Process

The Collector process completes certain steps per phase. When the AWHOST file is created, all work files are deleted.

You can manipulate the work files (after a processing phase has completed) by specifying **Proposals** and your own custom programs. Work files and AWHOST file layouts are described in the Reference section.

**Note:** Each workfile contains elements of the resulting AWHOST file.

Collector Processing phases are the:

- Prepare phase,
- Extract phase,
- Combine phase,
- Collect phase.

## Influencing the LegaSuite 5250 Collector Processing Phase

The Collector process can additionally be influenced during processing. Program flow can be interrupted to apply a proposal to your application collection.

To use proposals, you must know about RPG programming and details concerning your host application screen conditions. Influence the Collector to adjust or drastically change your AWHOST file contents.

Also, host application updates usually require re-running the Collector. In such a case, you can elect to rebuild the entire application or specify (by date) to collect on recently changed objects. In either case, you will need to transfer the AWHOST file to the Builder again. Keep a copy of your AWHOST files used for development.

## About the AWHOST File

A closer look at the Collector process begins by looking at the resultant file, the AWHOST. The AWHOST file contains all application display details. You must create this file on your host application using the Collector.

To do so, you must know details about the host application and associated libraries.

Even if you only have the application objects and not the source members, the Collector can scan the objects and gather enough information to make a set of application screens into an AWHOST file.

## Starting the LegaSuite 5250 Collector

Perform the following steps to start the LegaSuite 5250 Collector.

1. After you verify your application libraries, start the Collector.
2. Specify the application libraries and any changes to your **Current Library**.
3. Specify the **User Options Parameters** you require, and launch a batch job by pressing **Enter**. The Collector processes the application libraries in a phased approach.

4. Each phase achieves a result which is written to the allocated work file.

# Collector Prepare Phase

During preparation, the Collector tries to link the DSPF objects with their DSPF source members. Both DSPFobjects and their source members must be present within the library list in order for a link to be established.

If you plan to run the **LegaSuite 5250 Collector** using yourDSPF objects, then none of the DSPF objects should be in use. If an object is in use, the Collector will generate a warning message in the job log file and skip processing this object. The cause is most likely someone accessing the object (recordlock/ memberlock).

The Collector can automatically resolve some linking problems logged during the prepare phase. Others may require your intervention. Refer to the Introduction in the Errors and Diagnostic messages chapter for more details on **Prepare phase** messages found in your AWPRT spool files.

**Related topics**

Linking Objects and Source Members

## Linking Objects and Source Members

The prepare phase links objects and source members by performing the following steps:

1. Create a list of all sources in the library list.
2. Create cross references of all compiled objects and corresponding source members.
3. Allocate work files to be used in the extract phase.

**Related topics**

Collector Prepare Phase

# Collector Extract Phase

The **Prepare Phas**e linked all necessary files for collecting.

The **Extract Phase** reads all fields, texts and other relevant DDS keywords from DSPF (and MNU) source members and objects. The results are written to work the files temporarily allocated in your current library. The Collector may encounter DSPF source members having duplicate or non-ascending sequence numbers. Such a situation cannot occur in the OS/400 editor. It may be caused by source generating programs or by tools like DFU or SQL. The Collector can process these source members, but will still report this situation.

The **Extract Phase** may ignore a DSPF (MNU or Object) source. All ignored source files (DSPFs, etc.) will be reported in a spool file. The Collector is not capable of handling these objects in a correct way. It ignores DSPFs in the following cases:

1. The source member is of a type DSPF36 . Due to a different syntax, the Collector is not able to extract screens. You can convert DSPF36 source to "native" by compiling the source twice.
2. The object is of type DSPF38.
3. TheDSPF was found before in a source file already processed. The Collector will assume the first DSPF found to be the most recent one and will omit the latter one.
4. The associated object for the source member to be processed can not be located.
5. The associated object is in use and can not be processed.

6. The source member contains deleted records. Such a situation can not occur in the OS/400 editor. It may be caused by source generating programs or by tools like `DFU` or `SQL`. The Collector is not capable of handling these source members in a correct way.
7. The source member does not contain records at all. The Collector will not handle these source members.

# Collector Combine Phase

Each display file can be viewed as made up of screen formats. The Collector identifies these formats during processing.

Each recognized screen format (collected) can produce a unique screen reference in the `AWHOST` file. At the Combine phase, the Collector will try to combine record formats (specified by Keyword). The combination of common screen formats reduces the number of actual screen references included in the `AWHOST` file. A combination will be regarded as a new format and can be combined with additional formats under the same conditions described below.

The combination of formats requires these conditions be observed:

1. Formats must be within the same display file.
2. The last row occupied by the first format must not overlap the first row of the second format.
3. The OVERLAYkeyword or PUTOVRkeyword must be defined in at least one of the two formats to be combined.

## Uncombined Formats

The following formats will not be combined unless otherwise noted:

- Formats containing the WINDOW keyword will be regarded as non-combinable. However, formats referencing the same window may be combined, according to the rules described above.
- All formats containing the SLNO(*VAR)keyword will be regarded as non combinable.
- All formats containing the CLRLkeyword will be regarded as non combinable. This is untrue when all rows from the top row of the format up to the bottom row of the screen (normally 24) are cleared. You may, however, consider to ignore the `CLRL` keyword in the combine phase. Refer to User Options Parameters for a detailed explanation.

## Collector Work Files

The Collector stores processing results into work files. Work files are touched based on the processing phase. Work files are always allocated in the current library. This is to enable processing of multiple host applications at the same time. Use a different current library for every package you want to process.

The Collector creates a series of work files during processing (see Introduction in the Work Files layout chapter for the details to these files). **Note:** Upon completion of Collector processing, all work files are removed. Proposals and your own user defined programs can be executed after the [indicated] phase(s).

Work files and the associated processing phase are listed below.

*Table 1. Work Files and Associated Processing Phases*

| File | Description | Processing Phase |
|---|---|---|

| | | |
|---|---|---|
| AWOBJD | Contains a description of all objects involved. | [PREPARE] |
| AWMB | Contains a list of all file members found in the library list. | [PREPARE] |
| AWFMT | Contains a list of all formats found in all display source member | [EXTRACT,COMBINE] |
| AWTXT | Contains a list of all fields and texts extracted from all display source members. | [EXTRACT,COMBINE] |
| AWDFM | Formats not to include in final result . | [COLLECT] |
| AWSBF | All subfiles specified. | [EXTRACT/COMBINE] |
| AWWDWA | All windows attributes specified . | [EXTRACT/COMBINE |
| AWWDWB | All windows specified. | [EXTRACT/COMBINE] |
| USREDT | Show edited lengths for fields using special edit code 5–9. | [EXTRACT/COMBINE] |
| AWHOST | Final result, to be sent to the PC. | [COLLECT] |

**Related topics**

Appendix D : Collector Work File Layouts

# Collector Extract Phase

The **Prepare Phas**e linked all necessary files for collecting.

The **Extract Phase** reads all fields, texts and other relevant DDS keywords from DSPF (and MNU) source members and objects. The results are written to work the files temporarily allocated in your current library. The Collector may encounter DSPF source members having duplicate or non-ascending sequence numbers. Such a situation cannot occur in the OS/400 editor. It may be caused by source generating programs or by tools like DFU or SQL. The Collector can process these source members, but will still report this situation.

The **Extract Phase** may ignore a DSPF (MNU or Object) source. All ignored source files (DSPFs, etc.) will be reported in a spool file. The Collector is not capable of handling these objects in a correct way. It ignores DSPFs in the following cases:

1. The source member is of a type DSPF36 . Due to a different syntax, the Collector is not able to extract screens. You can convert DSPF36 source to "native" by compiling the source twice.
2. The object is of type DSPF38.
3. TheDSPF was found before in a source file already processed. The Collector will assume the first DSPF found to be the most recent one and will omit the latter one.
4. The associated object for the source member to be processed can not be located.
5. The associated object is in use and can not be processed.
6. The source member contains deleted records. Such a situation can not occur in the OS/400 editor. It may be caused by source generating programs or by tools like DFU or SQL. The Collector is not capable of handling these source members in a correct way.
7. The source member does not contain records at all. The Collector will not handle these source members.

# Collector Collect Phase

The Collect phase produces the final results of the completed process (`AWHOST` file) to reside in the current library.

This file should be transferred to PC, and will be the starting point for the Builder. Finally , when this phase is completed, all work files in the current library are removed.

# Chapter 4: Summary — Steps to Collect the AWHOST

Here is a summary of the steps you will perform to produce an AWHOST file.

General steps to use the Collector to produce the AWHOST file are as follows.

1. Identify the Host Application Libraries to Collect.
2. Start the Collector program and verify your current library and library list .
3. Specify **User Options** to apply and start the Collector job.
4. Review Collector results on the host application.
5. Transfer the AWHOST file to a LegaSuite Builder.
6. Review the Picture Album on your PC.
7. If too many unidentified screens are produced in the Service Builder, specify **User Options** and/or **Proposals** to influence the Collector process with your additional processes. When you have prepared and specified any **Proposals** (or other options), re-run the Collector process.

## Identifying the Application Libraries for Collection

The result of this step can be entered onto the Collector interface (edit library list). The Collector processes source members including referenced user message files (MSGFs).

The library list will normally be equal to the library list used when running the package. In addition, the libraries with the source members should be added to the list. The order of libraries in the Collector library list should be the same order as the library list used within your application.

### Library Identification Issues

When identifying the library, note the following **library list** issues:

- Normally, there is a one-to-one relationship between DSPF objects and members. If there are multiple source members for an object, or multiple objects with the same name, the first object and source member are used.
- If the host application uses message files, make sure the message files can be found in the library list.
- The Collector must have *USE authority for all libraries defined within the library list.

### Collector Preparation for Application Libraries

Before starting the Collector, you will need to check the following issues for the library list.:

1. Find out which libraries on your host actually contain source members with screens (DSPF and MNUs) to be processed.
2. Find out where the compiled objects reside.

# Starting the Collector and Verifying the Libraries

This activity is performed from the Collector Main screen. The current library will hold all interim working files and the created AWHOST file. You can set the library before using the Collector by changing the current library using **CHGCURLIB**. It is important to ensure that you have "write" authorization to the library you will be working within, otherwise the Collector will terminate with an error message.

Perform the following steps to start the Collector:

1. You can start the Collector with the command: `SEACOL/COLLEC`. At the copyright screen, press **Enter** and the main screen of the Collector is displayed. It displays the library list and current library.
2. Press **F16** to update the library list if necessary. The library list should contain all of the Application Libraries.
3. Press **F17** to update the current library if necessary.

## Altering the User Options

**User Options** influence the Collector operation.

Perform the following step to alter the User Options:

1. From the **Collector Main Screen**, check the user options by pressing **F18** and traversing through the menu options.
2. A menu of different options will appear. Navigate through the options by entering the number beside the option on the command line and pressing **Enter.**

   User options include:

   ```
   Parameters
   Collector Program Flow
   Indicator settings
   Formats to remove or omit
   Language specific options
   ```
3. Select and specify each option for your Collection Run.
4. Press **Enter** to return to the Main Screen.

As you perform the Collector operations, it may be necessary to specify *additional* User Options.

1. Update your host application Collector User Options as required.
2. When the Builder produces limited screen identification (your Picture Album contains too many unidentified screens?) you may need to affect the application Collector with additional **User Option Parameters** and/ or **Program flow** specification

## Starting the Collector Run

Once the environment has been set up correctly, you can start the Collector run.

1. Press **ENTER** on the Collector main screen.

   This submits a batch job called COLLECTOR.

If you run the Collector over objects that are in use (someone is running the application being collected), you will receive a warning message in the job log informing you that the object could not be collected.

2. Once the COLLECTOR job finishes, check the collector results on the host application by first evaluating the job log and spool files.
3. Check your library for the presence of a newly created AWHOST file. If no AWHOST file is created, there will be remaining work files containing the current results of the process.
4. The next step is to transfer the AWHOST file to a PC.

# Transferring the AWHOST File

The transfer of the AWHOST file from the host application to the PC can be done using any software that provides communication between the host application and PC.

However, before carrying out the transfer, the following steps need to be performed.

1. You should check the AWPRT spool files in the SEACOL/AWOUTQ to see whether any serious problems were encountered during the process. Review each message and appropriate actions. Example:

```
WRKSPLF  Check for user data AW310
WRKSPLF  Check for user data AW810.
```

2. Investigate the job log (WRKSBMJOB). If there are serious problems, then an AWHOST file may not have been generated.
3. If you correct any identified problem(s), you must specify those corrections (User Options) and restart the process.
4. When the Collector run is successful, the next step is to transfer the AWHOST file.

**Related topics**

Transferring Files Using Shared Folders

## Transferring Files Using Shared Folders

The transfer of the AWHOST file from the host application to the PC can be done using any software that provides communication between the host application and PC. When transferring AWHOST to the PC, the file should be transferred as an ASCII text file. A suggested technique is to transfer the files using the **Shared Folders** function.

1. To transfer files through the Shared Folders function, you must have a communications package which supports the Shared Folders function, for instance **Client Access.**
2. Start Client Access.
3. Enable the **Shared Folders** function by entering the following command on the iSeries command line. CPYTOPCD FROMFILE(<YOUR_CURRENT_LIB/AWHOST>)TOFLR(TEMP) TRNTBL(*DFT) TRNFMT(TEXT)
4. Sign on to the host application.
5. Type CPYTOPCD and press **F4** .
6. Specify AWHOST at the **From file** prompt.
7. Specify the library you used as current library during the Collector run at the **Library** prompt.
8. Specify the folder (directory on your host application drive, e.g. I:) to receive the AWHOST file, e.g. ‘TEMP’.

9. Press **ENTER.**

   If the command completed successfully, the host application will respond with "File member copied to PC document AWHOST". You should be able to access the file with a path like `I:\TEMP\AWHOST`. You are now able to view the `AWHOST` file.

**Related topics**

Transferring the AWHOST File

# Evaluating the Created Picture Album from the AWHOST File

Once the `AWHOST` file is transferred to the PC, it is necessary that you build a Picture Album using the Builder. Please see the related links on reviewing the Overview section concerning influencing Collector results, Collector User Options and other methods of influencing the Collector.

Perform the following steps to do so:

1. Review the screens produced using the `AWHOST` file.
2. Run **Picture identification** in the Builder.

   After you have run Picture Identification, you may evaluate the usability of the current Picture Album (from the generated `AWHOST` file).
3. Usually, if you are not satisfied with the results of the Collector run, it is because there are too many unidentified screens from the Picture Album.

   Consider using the following methods when you come across the following Picture Album issues. Apply the appropriate method(s) and re-run the Collector.

   - If you have identical pictures except for the function keys listed at the bottom, apply **Combine Function Key Formats**
   - If the Collector run is producing too many pictures from individual record formats, apply **Combine Formats Using PRECOL.**
   - If record formats are not being combined correctly, producing pictures that only display the top half or bottom half of a screen, apply **Combine Formats Ignoring CLRL Instructions**.

**Related topics**

Influencing the Collector Results

# Influencing the Collector Results

The goal of the Collector is to create an optimal `AWHOST` file. The `AWHOST` file can be evaluated. When the evaluation results indicate a better `AWHOST` file could be generated, you should produce a new `AWHOST` file by influencing the Collector and re-running.

Influencing the Collector means:

- Influencing the Collector with User Options.
- Applying Proposals and User-defined programs as they influence Collector Results (work files and AWHOST).
- Specify Indicator Settings Options,
- Removing or Omitting Formats,
- Specifying Language Options.

The Collector provides the following **tools** to influence the Collector results:

- User Options,
- Program Flow,
- Several other parameters for influencing the result.

The Collector job log and processing messages provide "hints" as to how the Collector results occurred.

**Related topics**

Evaluating the Created Picture Album from the AWHOST File

# Collector User Options

The greater part of control over the Collector process is provided within the **User Options** menu. **User Options** span the operational parameters to use the Collector, as well as specific selections related to specific host application environments (display methods, language modes, etc.).

User Options include:

- Parameters,
- Program Flow,
- Indicator Settings,
- Formats to Remove or Omit,
- Select Language of Collector Screen,
- User Program Flow and the use of Proposals are also highlighted .

**Related topics**

Appendix A: User Options Parameters

# Chapter 5: Collector User Options

The greater part of control over the Collector process is provided within the **User Options** menu. **User Options** span the operational parameters to use the Collector, as well as specific selections related to specific host application environments (display methods, language modes, etc.).

User Options include:

- Parameters,
- Program Flow,
- Indicator Settings,
- Formats to Remove or Omit,
- Select Language of Collector Screen,
- User Program Flow and the use of Proposals are also highlighted .

**Related topics**

Appendix A: User Options Parameters

## User Options Parameters

The Collector provides various parameter options when controlling the effect of combining record formats.

In evaluating your Collector run, if a large number of unidentified screens exist within your Picture Album (on the Builder), one method of reducing the screens produced is to combine screen formats. You can combine several types of screen formats. Screen formats containing function keys can be combined. Screen formats including the CLRLkeyword may be combined.

**Related topics**

Appendix A: User Options Parameters

## User Options — Collector User Option Parameters

Using parameters, you may affect the way the Collector reacts to commonly used display programming methods and control the resulting output.

To reach the **User Options** page perform the following steps:

1. Start the Collector with the command: `SEACOL/COLLEC`
2. From the Collector Main screen, specify **F18 User options** and enter a `<1>` for **Parameters**.

**Related topics**

Appendix A: User Options Parameters

Using Proposals to Combine Formats Properly

## User Options — Collector Program Flow

The LegaSuite 5250 Collector will allow you to set programs, which will be executed during the Collector process. This will make the LegaSuite 5250 Collector open to whatever program you want to be included in its process.

When you are using the **Collector Program Flow**, be aware that control of the data is passed to one or more user defined programs. Results can therefore no longer be guaranteed and are the responsibility of the user. The following table indicates the **Program Flow** options and explanation per option.

- **User Defined Program After Prepare Phase.** The prepare phase results in two files residing in the current library and in a spool file. The data files are AWMB (containing all members found in the user part of the library list) and AWOBJD (containing all objects of type *DSPFfound in the user part of the library list). A common use of the Collector Program Flow at this stage would be to inspect the spool file (and take appropriate action) or to remove certain members from the AWMB file, or to remove certain objects from the AWOBJD file.
- **User Defined Program After Extract Phase.** The extract phase results in two files, both residing in the current library. The data files are AWFMT (containing all separate formats found) and AWTXT (containing all texts and fields found). At this stage, the **Collector Program Flow** can be used to create your own screen combinations or add some conditional text to the picture formats.
- **User Defined Program After Combine Phase.** The extract phase results in two files, both residing in the current library. The data files are AWFMT (still containing all separate formats, but also containing the combinations created by the standard procedure or by the "user defined program after extract") and AWTXT (still containing all texts and fields found). After this stage, the Collector Program Flow can be inspected for all screen combinations in order to add, change or delete them from the AWHOST .
- **User Defined Program after Collect Phase.** The collect phase results in the creation of the AWHOST file. At this stage, it is possible to directly manipulate the contents of the AWHOST file.

**Related topics**

Influencing the Collector Program Flow

## Influencing the Collector Program Flow

Perform the following steps to influence the **Collector Program Flow:**

1. From the **Collector Main screen,** specify **F18 User Options** and enter a <2>to display the **Program Flow screen**.
2. Specify a proposal or user program to invoke by specifying the program name in the field associated with a process phase (invoked after phase completes).
3. The interim results of the four phases of the Collector run (stored in working files) can be influenced by user programs. In between phases, user-defined programs can modify the Collector process by performing additional manipulations after the specified process phase. Use Proposals (or copy a proposal as a base from which you can create the customized values within program) and/ or user-defined programs to manipulate the work files (and resulting AWHOST file).

**Related topics**

User Options — Collector Program Flow

# User Options — Collection of Language Display by Indicator Settings

Some applications make heavy use of indicators to implement multiple languages in one display file. In these situations every text is in the DSPF multiple times in different languages and dependent on some specific indicators.

The resulting picture contains all the texts and will display them in the sequence of statement numbers in reverse order. If the language is not always the last statement then the picture will display

the mixed language! Using this option you are able to control which language (indicator setting) will display in the Picture Album.

Use the **Language Display** screen to select fields and texts, that are conditional based on indicators. Texts and fields are examined for the use of indicators. If the marked indicator is encountered (marked as 0 if (Nxx), marked as 1 if ( xx)) the text will be marked in such a way that it will appear when examining the pictures on PC.

Suppose an example source (generated by the IDDOS tool) looks like:

```
0001.00 A N61N62 63
0002.00 AAN64N65N66    row col 'GERMAN TEXT'
0003.00 A N61N62N63
0004.00 AA 64N65N66    row col 'ENGLISH TEXT'
0005.00 A N61N62N63
0006.00 AAN64N65 66    row col 'DUTCH TEXT'
```

Marking indicator **63** as 1 would select the German text, thus showing the German text when looking at pictures on PC. Although the English and Dutch text are in the picture as well, they are "hidden" under the German text. To choose the English text, indicator **64** would be marked with a **1**. To choose the Dutch text, the indicator **66** would be marked with a **1**.

# User Options — Collector Select Language Option

The Collector uses message files for its texts. You may make a copy of an existing message file and translate the contents into your native language. So your users will operate the Collector in their own language. If the message is in the *SEACOL* library, you can activate the message file using this option. If you are using Collector on a DBCS version of iSeries then your Collector language is automatically set to uppercase English.

# User Options — Collection Formats to Remove or Omit

Using this option, record formats can be marked by their name.

Some applications have one or more "special" formats copied or generated in display files, like `DUMMY` or `CLEAR`. Another common way of coding record formats is to include a format in each `DSPF`, showing for instance the copyright of the package. The result could be that the Collector generates many of the same looking pictures or too many format combinations. You may decide to omit these record formats in the Collector run, which will leave you with less pictures.

This option is used to exclude those formats by name from the Collector process.

You could mark these record format names as:

- **<1>**. Specifies the Collector to ignore all occurrences of this record format.
- **<2>**. You may decide to include these record formats in the Collector run only once, and omit all other occurrences by marking these record format names as "1".
- **<3>**. Finally, there may be record formats, which will not be combined with any other record format at runtime, even though the source would suggest so. You could mark these record format names as <2>, which will result in a separate picture for this record format. Typically this would be the case for the record formats, looking like windows, but coded with reversed blanks to show the borders instead of the WINDOW and WDWBORDER keyword.

# User Options — Collector Select Language Option

The Collector uses message files for its texts. You may make a copy of an existing message file and translate the contents into your native language. So your users will operate the Collector in their own language. If the message is in the *SEACOL* library, you can activate the message file using this option. If you are using Collector on a DBCS version of iSeries then your Collector language is automatically set to uppercase English.

# Chapter 6: Using Proposals to Influence the Collector

Proposals affect the results contained in the AWHOST file. Proposals influence the Collector results by modifying the interim work files produced after each phase of the Collector run. Consequently, it is necessary that you are familiar with RPG, the iSeries development environment and your own application in order to make use of proposals.

Collector proposals affect AWHOST results relating to:

- Handling of subfiles and attributes,
- Collecting of code generated display files (DSPF),
- Listing specific source members to collect,
- Halting the collector during processing,
- Handling of specified record formats (for removal or combination of pictures),
- Launching of two proposals during a processing phase.

Proposals are described and contained within the **SEACOL/PROPOSALS** source file. Use these proposals as example programs to copy and change based on your needs. Keep in mind that these Proposal programs may change in future releases of the Collector.

**Note**: A complete description of the purpose of each proposal can also be found in the associated source member.

## Collector Proposal Descriptions

Proposals listed in the **Appendix** chapter are grouped according to an iSeries "element" of the display files which you want to affect during the Collector process. A complete description can be found in the associated source member.

**Note:** These proposals should be viewed as example programs you can model your needs around. RPG knowledge is essential to actually inspect and manipulate these proposals into your own Collector programs. Each description includes the proposal name, program type, and at which process phase it should be executed

Proposals can be used for changing attributes, etc. at the field-level. Examine these proposals when interested in changing field (text-based) attributes such static text into messages; messages into static text, and so on.

**Related topics**

Appendix B : Collector Proposal Descriptions

## Accessing Collector Proposals

Proposals enable you to influence the results of the Collector. Proposals are example programs that you can customize and add to the Collector program flow.

Perform the following steps to access Collector proposals.

1. From the Collector main screen, choose to access **User options (F18).**

2. Select the **Collector program flow menu** option. A dialog will be displayed which enables you to enter the name of a program (and the library) to be executed after the completion of a Collector phase.

**Related topics**

Using Proposals to Combine Formats Properly

# Chapter 7: Collector DDS Keyword Handling

**DDS Keyword**s describe display conditions within your `DSPF` and `MNU` files. The Collector examines supported DDS Keywords and can perform operations based on their presence. Relevant elements are extracted from DSPF's with the type `DSPF`and `MNUDDS`and from source members with the type `DSPF38`.

Fields and texts are extracted from the members (or objects) that are found. All source statements are scanned to find information, like screen positions, field attributes, conditions and other relevant information. Fields and texts are gathered for each record format. The `SFL` record format and its associated `SFLCTL`record format are considered to be one format. All keywords are examined.

The Collector will combine two record formats, when the formats contain the OVERLAYor PUTOVR keyword.

**Note:** Overlapping formats have restrictions when combining them.

For a detailed description of all possible keywords, see the "Data Description Specifications Reference" (SC41-9620 -00). A list of DDS keywords supported (recognized) by the Collector is included in the **Appendix — Host Application DDS Keyword Listing** chapter.

**Related topics**

Appendix C : Host Application DDS Keyword Listing

# Chapter 8: Collector Errors and Diagnostic Messages

This chapter contains information pertaining to the error and diagnostic messages which will be reported in the `AWPRT` spool files in the `SEACOL/AWOUTQ`. Always check the spool files for any messages after launching the process. Messages are grouped by :

- The **Prepare Phase.** The Prepare phase occurs first.
- The **Extract Phase** (in which they occur in the collection process).
- The **Extract phase** occurs after the Prepare phase.

**Related topics**

Appendix E : Collector Messages

# Chapter 9: Troubleshooting Assistance

The following topics provide troubleshooting assistance should you come across Collector issues.

## Combining Function Key Formats

If you do not get the results you were expecting when you run the Collector and create a Picture Album, you may need to use one or more proposals to influence the way the Collector gathers information. If you have multiple function key formats in the footer of many screens, you may get a different picture for each function key format.

The screenshot below shows an example.

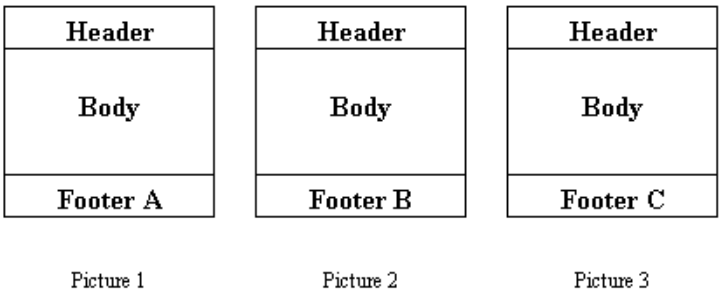



*Figure 1: Combining function key formats example*

The desired result is a single picture that contains all the function key formats. You can set a parameter that tells the Collector which row your function key formats start on, and the parameter will combine all formats into one picture.

For example, if your function key formats start on row 22, perform the following steps to combine the function keys:

1. On the Collector's **Main Menu** screen, type `SEACOL/COLLEC` on the command line.
   You are brought to the **Welcome To LegaSuite 5250 Collector** screen.
2. Press **Enter** on the **Welcome** screen.
   You are brought to the **LegaSuite 5250 Collector Main Screen.**
3. Press **F18** .
   You are brought to the **LegaSuite 5250 Collector User Options** screen.
4. Choose option 1, **Parameters.**
   You are brought to the **LegaSuite 5250 Collector User Options** – parameters screen.
5. Enter the row number “22” in this field that starts with **“Combine record formats starting from row”**

6. Run the Collector.
   You should have one picture containing all three footer formats.

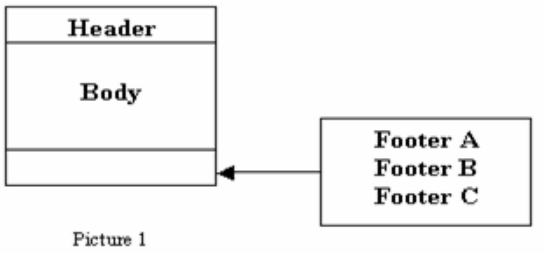



*Figure 2 : Result of combined function key formats*

# Combining Consecutive Single Row Formats

If your application uses single row formats that make up a display file, you can have the Collector combine those formats into one format to produce fewer, more accurate pictures.

A single row format is combined with another if:

- It only occupies one row, and
- The next format occupies one row as well and is equal to or immediately following the row of the first format.
- If your single row formats are used to simulate the appearance of a subfile, you can create a subfile effect using this option.

Perform the following steps to combine consecutive single row formats:

1. On the Collector's **Main Menu** screen, type `SEACOL/COLLEC` on the command line.
   You are brought to the **Welcome To LegaSuite 5250 Collector** screen.
2. Press **Enter** on the **Welcome** screen.
   You are brought to the **LegaSuite 5250 Collector Main Screen.**
3. Press **F18.**
   You are brought to the **LegaSuite 5250 Collector User Options** screen.
4. In the Collector's User Options menu, select the **Parameters** option, and in the parameter field that starts with **"Combine consecutive single row formats"**, type `1` in this field.
5. Another way to combine single row formats is by using a proposal. Here are the proposals you can use to force the Collector to combine consecutive single row formats:

| **Use this proposal** | **To have the Collector combine single row formats...** |
|---|---|
| APPL013 | Row by row |
| APPL032 | Two rows by two rows |
| APPL033 | Three rows by three rows |
| APPL034 | For rows by four rows |

Each of the above proposals compares the consecutive rows, and if they are equal, it combines them.

# Using Proposals to Combine Formats Properly

For non-standard applications, the Collector includes programs (called *proposals*) you can use to control the results of the collection. The source is included for each program so you can edit them to suit your needs if necessary.

You specify what proposals you want to use in the **User Options** before running the Collector, using the **Collector program flow** option. It is important that you apply a proposal after the proper phase of the collection. Within the source of each proposal, you can read a detailed explanation of its purpose and the proper Collector phase to apply it.

**Related topics**

Accessing Collector Proposals

User Options — Collector User Option Parameters

# Using the PRECOL Proposal

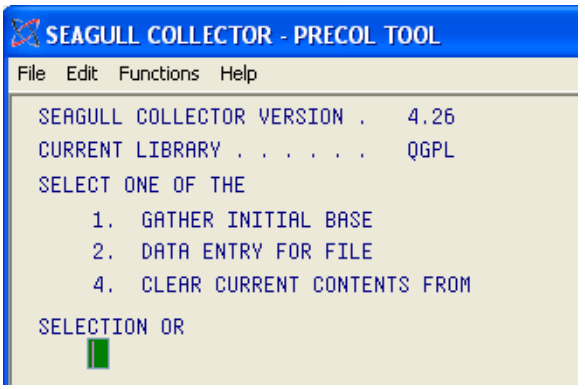
Using several PRECOL proposals, you can control how formats are combined in one display file or across several display files to create screens and panels. Before you undertake this procedure, you should understand your display files and the names of their formats.

Perform the following steps to use the PRECOL Proposal.

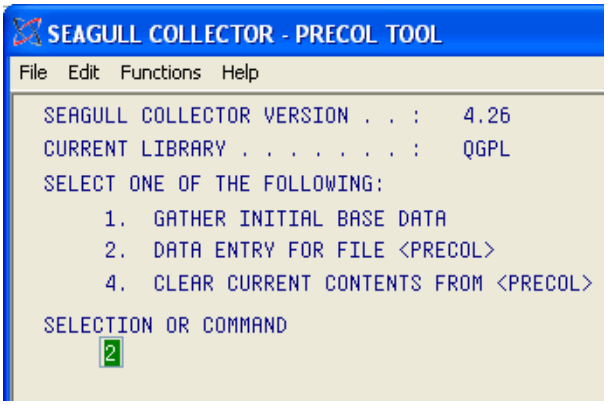
1. On the Collector's **Main Menu** screen, type `SEACOL/COLLEC` on the command line.
   You are brought to the **Welcome To LegaSuite 5250 Collector** screen.
2. Press **Enter** on the **Welcome** screen.
   You are brought to the **LegaSuite 5250 Collector Main Screen.**
3. Press **F18** .
   You are brought to the **Legasuite 5250 Collector User Options** screen.
4. Click on the **Collector Program Flow** option.
   The Collector Program Flow screen is displayed.
5. In the screen, go to the **Combine Phase.** Run the Collector using the MAKEPRECOL proposal after the Combine phase by typing `"MAKEPRECOL"` in the field after the User Defined Program after Combine field.
6. In the same **Combine phase**, type in `"SEACOL"` in the field after the **Library** field.
   This proposal creates two files on the iSeries that you can edit to specify how formats should be combined:
   - `PRECOLNW` is a data file that allows you to combine formats within the same display file.
   - `PRECOL60` is a logical file – related to the `PRECOLNW` file – that allows you to combine formats across multiple display files.
7. Exit the **Program Flow** screen and exit the Collector until you return to the command line.

8. At the command line, enter `SEACOL/PRECOLTOOL` on a command line to open **PRECOLTOOL**, a file editor program.

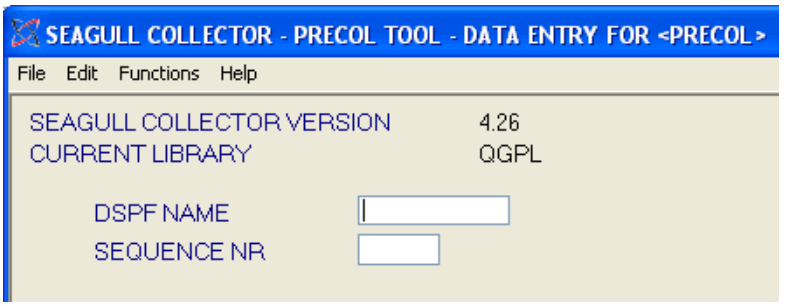
   The **PRECOL** tool menu is displayed.




9. Type **1** and press **Enter** to view the data gathered about the display files you collected against. You need to do this the first time you use PRECOLTOOL. After that, you do not need to run it again as long as your display files do not change.
10. Edit the library list to include the libraries on which you ran the Collector.
    a. Use **F19 (Edit library list)** to retrieve the list you used in the Collector run.
    b. Press **Enter** to send **PRETLBAS** to batch.
    c. Exit the Collector to return to the command line.
11. Type `SEACOL/PRECOLTOOL` on a command line again, then press **Enter.**
    a. At this screen, type **2** and press **Enter** to edit the PRECOLNW file.



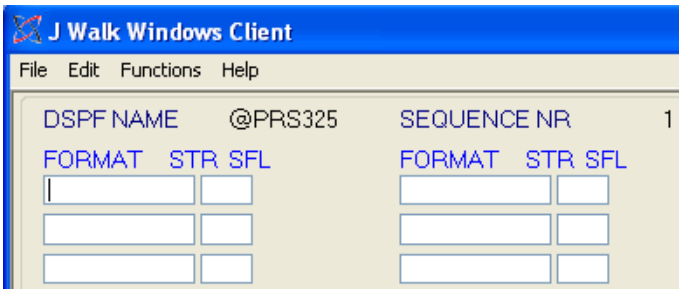

    b. Press **F4** in the **DSPF name** field to select the display file you want to use.




    c. In the displayed popup that lists the display files, type **1** next to the display file and press **Enter.**

       The **DSPF name** and **Sequence number** fields are filled with the selection you made from the display file list pop-up.

12. You now have two options:
    - Press **F9** to retrieve a new sequence number and press **F6** to display a dialog that allows you to enter a new format combination for this display file.




    - To edit an existing combination, enter the sequence number previously assigned for a combination in this display file and press **Enter.**
13. If you go with the first option (pressing **F9** ), press **F4** in the first field of the Format column to display a dialog with a list of formats for this display file.
    a. In the dialog, type a number (1 through 540) in front of each format you want to combine to specify the order in which they should be combined. The formats are listed in the order you specified.

       For formats coded with the keyword SLNO(*VAR), you can specify the row where you want the format to start in the **Str** column. This provides a method of including such formats on more than one row within your picture, in the same way as the application program.

    b. Press **Enter** to save this combination and press **F6** to add a new combination for this display file, or specify a different display file in the **Display name** field and then press **F6.** Repeat the steps for combining formats.
14. After you have specified all of your combinations, run the Collector again, this time using the **APPL061** proposal after the **Extract phase.** The **APPL061** proposal uses the `PRECOLNW/PRECOL60` file to combine formats.
15. Create a Picture Album from the new `AWHOST` and check to see that your formats were combined correctly.

# Eliminating Display Files from the Collection

Using the `PRECOLNW` file, you can specify display files to omit from the collection process.

Perform the following steps to eliminate display files from the collection:

1. Use **PRECOLTOOL** (explained in the topicUsing the PRECOL Proposal, on page 28 ) to edit the `PRECOLNW` file.
2. For the display file that you want to omit, enter NOTHING (or any other word that is not a format name) as the format to combine.
3. Then use the **APPL061** proposal when you run the Collector. This display file will be omitted because the format does not exist.

# Creating One Picture that Works for Multiple Languages

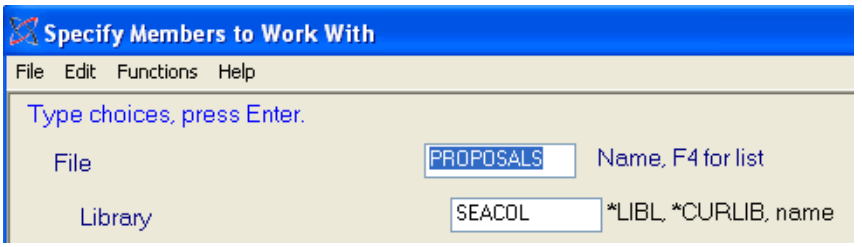
You can create one set of screens to handle multiple languages, as long as you have equal sets of display files for each language. The contents of the literal will of course vary among languages, but the length of the literal should match for each language. (You can use the **Text Translation Tool –TTT–**

to achieve this.) You can create the AWHOST file from just one of the sets of display files (for one of the languages). You should specify that literals be regarded as messages.

To do this, use the **LIKEMSG** proposal after the **Combine phase** when running the Collector. The proposal assumes that fixed text should be regarded as a message. Then as long as your text panel fields are set to Always update text, one panel can handle all versions of the text message. **Note:** By default, the proposal treats the whole screen. You can edit the program so that it begins and ends on certain rows.

Perform the following steps to create one picture that works for multiple languages.

1. On the host, type STRPDM on a command line and press **Enter** to start the **Program Development Manager.**
   a. Choose option 3, **Work with Members.**




   b. Enter PROPOSALS as the file and SEACOL as the library.
2. Find the **LIKEMSG** proposal, then type 2 next to it and press **Enter.**
3. Page down to the **Default Values for Parameters** section.

   Under ***LIKE DEFN AWLINE STRROW**, notice the Z-ADD1 parameter. The **1** specifies the row number on which the proposal will start.
   a. Change the **1** to the row number on which you want the proposal to start.
   b. Under ***LIKE DEFN AWLINE ENDROW**, change the **Z-ADD** parameter to specify the row number on which the proposal will stop.
   c. Press **F3** . On the **Exit** screen, leave **Y** in the **Change/create member field** and press **Enter** to save your changes. A message states that **LIKEMSG** has been changed.
4. Once you have changed the source, you need to compile it. Before you compile, make sure that **SEACOL** is part of your library list. To compile, type 14 next to the **LIKEMSG** member and press **Enter.**

   If you see the **Confirm Compile of Member** screen, type Y in the **Delete existing object** field to replace the current **LIKEMSG** program. (Depending on your host application settings, you may not see this screen.) The compile is sent to batch. (Depending on your host application settings, the compilation may be performed interactively.)
5. Add the **LIKEMSG** proposal to the Collector run after the Combine phase.

# Appendix A: User Options Parameters

The following list explains each of the user options as they appear on the **User Options – Parameters** screen.

**Combine record formats starting from row**

For each `DSPF`, the Collector finds all formats starting at or below the row indicated and regards them as a single format, regardless of any keyword specified. This can dramatically reduce the number of pictures that the Collector run makes. Use this parameter to combine formats containing function key references (e.g. row 22 or 23) and program messages (row 24) into one format. This new format can then be combined to other formats under the conditions described on the combine phase. Formats combined in this way, can be recognized by their name, which will be *0XX/0YY, where XX stands for the value you have entered and YY stands for the display height, normally 24. The default value is <0>, which indicates that this parameter will not be used

**Combining Function Key Formats**

Use this **User Options** method when you have pictures being produced that are identical except for the function key assignments displayed at the bottom of the screen. This commonly occurs when there are more function keys available than can be displayed on one screen. This solution combines these pictures into one and leaves the last lines of the screen variable. Follow these steps to perform this activity.

1. After pressing **F18** to display the **User options** menu, choose menu option **1** to display the **Parameters** screen. The cursor will be positioned at the first parameter, **Combine record formats** starting from row. By default, this row number is set to **<0>** which indicates that all rows should be used when generating record formats.
2. Change the row number to the first row where function key assignments are displayed at the bottom of the screen. This is usually at row 22. Now when the Collector run is started, the function key rows will not be considered unique record formats and only one picture will be generated, instead of one picture for each of the different function key assignment record formats

.

**Combining Screen Formats Using PRECOL**

If there is more than one format used to describe a screen and there are multiple screens in one source, all formats will be combined. This can form more pictures than the original number of screens. Use this method when record formats are being combined into too many pictures. The information is presented in an example format. If you have two screens within one `DSPF`, each made up of a header, body and footer, the screen permutations of this combination form eight different picture possibilities.

1. HEADER1+BODY1+FOOTER1
2. HEADER1+BODY1+FOOTER2
3. HEADER1+BODY2+FOOTER1
4. HEADER1+BODY2+FOOTER2
5. HEADER2+BODY1+FOOTER1
6. HEADER2+BODY1+FOOTER2
7. HEADER2+BODY2+FOOTER1
8. HEADER2+BODY2+FOOTER2

Suppose the only combinations that you want as pictures are combination 1 and combination 8. Enter the record formats at the end of the PRECOLproposal. Each picture will be defined using three rows, identifying the source, the picture record number, and the record format. The rows used to define two pictures, one for combination 1 and one for combination 8 would look like:

| Source | Pic # | Record Format |
|---|---|---|
| DSPF01 | 1 | HEADER1 |
| DSPF01 | 1 | BODY1 |
| DSPF01 | 1 | FOOTER1 |
| DSPF01 | 2 | HEADER2 |
| DSPF01 | 2 | BODY2 |
| DSPF01 | 2 | FOOTER2 |

The PRECOL file and PRECOLTOOL are provided to define specific screen identification formats without viewing the source code. The PRECOL file is used to hold the selected combinations of pictures (from Initial Base Data). The PRECOLTOOLmaintains the PRECOL file.

**Combine consecutive single row formats**

A package may be developed using a separate record format for each row displayed. In such cases the Collector has difficulties in combining the record formats. Not only would this take a considerable amount of time to process, the AWHOST file could contain a large amount of pictures "unknown" to the package at runtime. In such cases this parameter can help you save time, and still produce "known" pictures. Within a display file, record formats are sorted by their starting row. When the parameter has a value of <1>, record formats are combined when:

- The current record format only occupies one row and
- The next record format occupies one row as well and
- Both record formats are on the same row or on two consecutive rows.

**Omit empty record formats**

Setting this parameter value to **<1>** will cause the Collector to regard empty formats as non-existent. A format is considered empty when no fields or texts are displayed. Be aware that program and hidden fields as well as fields or texts with an unconditional ND (non-display) attribute are not displayed. (This is not true for message fields.)

**Omit records only containing blanks**

When this parameter is set to **<1>**, record formats only containing blanks will be ignored by the Collector. In most cases, record formats only containing blanks are used within a package to clear (part of) the screen and should be omitted. Keep in mind that this parameter is in effect for all display files of the package you are handling.

**Ignore <CLRL> keyword on combine phase**

The keyword is typically used in System 36 based packages. The OVERLAYkeyword is the corresponding iSeries keyword. When this format is sent to a screen, it doesn't clear all lines, but only from the first defined row until the last defined row in this format. The CLRL(*NO) keyword doesn't clear the screen. The CLRL(xx) keyword clears "xx" (where xx indicates a number of lines on the screen) number of rows starting from the first defined row of this format. If this parameter is set to <0>, combinations of record formats will be made under the following conditions:

- Record formats must be within the same display file, and therefore within the same source member.
- The last row occupied by the first format must not overlap the first row of the second format.
- The OVERLAYor PUT keywords must be defined in at least one of the two formats.
- If you set this parameter value greater than <0>, every format containing the CLRLkeyword is considered to have the OVERLAYkeyword defined.

**Note:** In order to combine formats, these formats should not be overlapping. The following values can be used for this parameter.

- **<1> Ignore both CLRL(*NO) and CLRL(xx)**. The Collector will try to combine these formats. The first row and the last row of these formats are as defined in the source.
- **<2> Ignore both CLRL(*NO) and CLRL(xx), do handle <xx>.** The Collector will try to combine these formats, but handle the format with CLRL(xx) starting from the first defined row and ending xx lines down.
- **<3> Ignore CLRL(*NO), do not combine CLRL(xx).** The Collector will try to combine all formats with CLRL(*NO), but won't combine formats with CLRL(xx). Every format gets a new picture.
- **<4> Do not combine CLRL(*NO), ignore CLRL(xx).** The Collector doesn't combine formats with **CLRL(*NO).** Every format gets a new picture. The Collector will try to combine all formats with CLRL(xx). The first row and the last row of these formats are as defined in the source.
- **<5> Do not combine CLRL(*NO), ignore CLRL(xx), do handle <xx>.** The Collector doesn't combine formats with CLRL(*NO). Every format gets a new picture. The Collector will try to combine format withCLRL(xx), but handle the format which using CLRL(xx) starting from the first defined row and ends xx lines down.

**Include PULLDOWN formats**

Use this setting to specifyPULLDOWN formats to be included in your pictures:

- <1>=Yes
- <0>=No.

**Use PRECOLTOOL to Create Pictures from DSPF Formats**

Use this tool to remove unwanted screen formats during the Collector run. Follow these steps to perform this activity.

1. Start the PRECOLTOOL by typing the command `CALL SEACOL/PRECOLTOOL`
2. To create pictures using formats from a display file (DSPF), you must have gathered initial base data. Specify `<1>`and press **ENTER.**
3. After you have gathered the initial base data is completed, restart the PRECOLTOOL and select the **“Data entry for file <PRECOL>”** function. Enter `<2>`and press **ENTER.**
4. You may select the **“Clear current contents of <PRECOL> file”** (option <4> from the menu) to essentially start again (or recover a very small amount of disk space.

**Note:** It may be more valuable to save the <PRECOL> file for reuse after the host updates have occurred.

**Gather Initial Base Data for PRECOLTOOL**

Before you can proceed using the **PRECOLTOOL**, you must gather initial base data. This function gathers information from the display files (`DSPF` s) for later use with the **PRECOLTOOL**. When you have completed this step and your display files have not changed (again), you can consider that the base data is gathered.

**Make PRECOL and AWHOST Files**

Begin the Collector session by specifying that the Collector process produce an `AWHOST` file and `PRECOL` file. Please refer to the member <MAKEPRECOL> in source file <SEACOL/PROPOSALS> for a detailed description of this entry. Follow these steps to perform this activity.

- From with the Collector Start screen, select **User Options (F18).**
- Call the **Collector Program Flow** entry (option <2>).
- Specify the program <MAKEPRECOL> in libray <SEACOL> within the **“User defined program after combine”** field. Results of this activity are stored in the PRECOLNW file.

**For OS/400 V2R3M0 or older**

This parameter must be set to **<0>** for DSPF38 objects, when using OS/400 V2R3M0 or lower and when using indicator settings. Refer to Indicator Settings for more information about using indicator settings to operate your AS/400 application in different languages.

**Reset Original Library List**

This parameter specifies whether or not the library list will be set to the original value at the end of the process.

**Run Process Interactive**

Setting this value to <1> will make the Collector run operate interactively. This is only done when debugging user programs that you have added to the Collector process in the Collector Program Flow.

**Prompt when submitting job <COLLECTOR>**

With the setting of this parameter you may indicate whether or not you want to change the default settings when submitting the job <COLLECTOR>. If you set this parameter to <0> (no, the job <COLLECTOR> will be submitted using the default values from your job description. You may want to change some of these settings; typically you may use this feature to schedule the job <COLLECTOR> to run in the absence of users. Changing the default settings may interfere with other jobs running on your system, which may be delayed by your interference. If you are in any doubt, contact your security officer before changing these settings.

**Only Handle Members Changed Since**

This parameter may be typically used to run the Collector on a new version of your package. By entering the date, only those DSPF's changed since the date entered will be treated. All other `DSPF` files will be omitted. A source member is considered changed when:

- One or more statements have been changed,
- The source member text has been changed or renamed

An object is considered changed when:

- The object is compiled or moved,
- The date should be entered in the format (YYMMDD), disregarding the status of your system value (QDATFMT). are in use. A warning is logged indicating processing has skipped an object currently in use.

**Collect on Object**

The Collector can extract information from `DSPF` objects without the corresponding source members. This parameter can only be used with OS/400 V3R1M0 (minimum). This parameter has three possible values:

- **<0> Extract information from source members only.** The source member must have a corresponding DSPF object.
- **<1> Extract information from DSPF object.** The object must have a corresponding source member.
- **<2> Extract information from DSPF object**. Existence of source member is not important.

It is recommended to set this parameter to <1> or <2> when possible because the Collector works faster with objects than with sources and objects reflect more accurately what fields appear on the screen at runtime than sources. Using this option (<1> or <2>) requires that none of the `DSPF` files are in use. A warning is logged indicating processing has skipped an object currently in use.

**Use CA/400 (W95 and NT) to Transfer AWHOST to PC (DBCS)**

Use this parameter when you use Client Access/400 (W95 and NT) to download the `AWHOST` file to your PC (<1>=Yes, <0>=No). This parameter is only available on DBCS systems. CA/400 requires

DBCS files (be transferred to a PC) to be created with the option IGCDTA(*YES). Other file transfer utilities DO NOT require use of this option. Incorrect use of this option can result in an `AWHOST`. The file (on the PC) with the incorrect "JIS"-code. The `AWHOST` will appear corrupt in the Builder.

**Add SO/SI To DBCS Graphic Text**

Set this parameter to **<1>** or **<0>** to add SO/SI (shift in/shift out) characters to DBCS Graphic (G-type) text to enable the DBCS graphic text to be seen as DBCS characters when the AWHOST file is transferred from the iSeries to the PC.

**Related topics**

User Options — Collector User Option Parameters

User Options Parameters

Collector User Options

# Appendix B : Collector Proposal Descriptions

All programs in the file <PROPOSALS> included with the Collector are listed below. .

## Changing Field Attributes

Proposals can be used for changing attributes, etc. at the field-level. Examine these proposals when interested in changing field (text-based) attributes such static text into messages; messages into static text, and so on.

**APPL011.Program Type – RPG. Process Phase – Extract.**

This proposal examines source members for output fields and replaces the output field with the related text. The name of the text file is hard coded to **<F9220>.** Output fields have names with the format VTXxxx and have a three-digit number, starting with 001. So, the first output field name is VTX001. This proposal represents a specific manipulation and will require modification for your use.

**LIKEMSG.Program Type – RPG. Process Phase – Combine.**

Use this proposal to specify all text fields are coded as messages <MSGID>. This proposal can be used to create one set of language independent panels (containing message references) which can refer to externalized, translated literal message files for use in supporting various national languages.

**SCREENID. Program Type – RPG. Process Phase – Combine.**

This program specifies screen title fields to become fixed text. Use this program to improve the results of screen identification, particularly when your application texts are coded as externalized <MSGIDs>. The program searches for each occurrence of the I/O field <##PGM>, , on row one, position two, and when found, the program specifies field to be fixed text (containing the first seven characters of the DSPF name. The converted field provides a unique identifier within your screens. This program can also assist in identifying applications created using SYNON development tools.

## Handling Subfiles and Attributes

Proposals can be applied to re-assign subfile attributes to field definitions under the appropriate circumstances.

**APPL031. Program Type – RPG. Process Phase – Extract.**

Rows that are coded as individual rows but appear as a subfile, have their attributes changed to that of a subfile in the resulting `AWHOST` file. There must be at least four rows of similar appearance.

**APPL032. Program Type – RPG. Process Phase – Extract.**

This program looks for sets of two rows that are similar in appearance and changes their attributes in the `AWHOST` file to that of a subfile with two rows per record. There must be at least four sets of rows of similar appearance.

**APPL033. Program Type – RPG. Process Phase – Extract.**

This program looks for sets of three rows that are similar in appearance and changes their attributes in the `AWHOST` file to that of a subfile with three rows per record. There must be at least three sets of rows of similar appearance.

**APPL034.** **`Program Type – RPG. Process Phase – Extract.`**

This program looks for sets of four rows that are similar in appearance and changes their attributes in the `AWHOST` file to that of a subfile with four rows per record. There must be at least three sets of rows of similar appearance.

## Collecting Code Generated Display Files DSPF

AS/400 display files can be created by "code generators." When such a tool is used to develop your AS/400 displays, certain deficits may need to be corrected before you run the Collector. Study these example proposals in order to see LANSA created display file deficits be corrected.

**LANSA.** **`Program Type – CLP. Process Phase – Extract.`**

This proposal reads DSPF sources that were generated with the LANSA code generator. All formats are then read and placed in the `PRECOL` file. This proposal was created for a specific development environment (I.e. LANSA) and is not suitable for general use.

**LANSA2.** **`Program Type – CLP. Process Phase – Extract.`**

This proposal reads the DSPF sources created by the LANSA code generator. This proposal is used in conjunction with LANSA. The first time that you execute the Collector process with LANSA DSPF sources, and the <LANSA> proposal to fill the `PRECOL` file. After that, always use the <LANSA2> proposal. This way, you can edit the `PRECOL` file and keep these combined format definitions.

**TESTDSPF.** **`Program Type – RPG. Process Phase – Prepare.`**

Tests if source members (with no type identifier) are of type <DSPF>. Occasionally, generated code is created without the required DSPF type identifier. Use this program to inspect (particularly LANSA generated code) display files (and when identified) set the type to "DSPF".

## Listing Specific Source members to Collect

When you want to select a specific set of source members for Collecting, you may submit a file (list) to limit the scope of the Collector run.

**APPL070.** **`Program Type – RPG. Process Phase – Prepare.`**

This proposal will only collect those members that are listed in the <MBRONLY> file. This file must be located in the library list or, preferably, in the working library. If the file <MBRONLY> is not found, the Collector process will halt with an error CPF4101.

## Halting the Collector during Processing

If you need to halt the Collector run (while investigating a proposal, etc.), apply the following proposal to do so.

## Handling Specified Record Formats for Removal or Combination of Pictures

**ERROR.** **`Program Type – RPG. Process Phase – Any phase.`**

This proposal temporarily halts the Collector process after the specified phase. To continue processing, enter "I". To stop processing, enter "C".

## Handling Specified Record Formats for Removal or Combination of Pictures

In the Collector process, record formats are a basis for identifying uniqueness amongst host application display files. In order to control record formats and how to handle them, study these proposals and the record format operations they perform.

**EQUAL10. Program Type — RPG. Process Phase — Extract.**

This proposal specifies record formats from the same display file (DSPF), where the six left-most characters are matching, to be grouped together in a single picture.

**APPL080. Program Type — RPG. Process Phase — Extract.**

This proposal removes formats found in the FMTREMOV file. This file must be located in the library list or, preferably, in the working library. If the file FMTREMOV is not found, the Collector process will end with an error CPF4101.

**APPL021. Program Type — RPG. Process Phase — Extract.**

Used in conjunction with <PRECOLNW> to combine record formats. Formats that start with the SLNO(*VAR) keyword are not combined and create their own record format. The last eight characters of the original record format name will be used as the name of the new record format.

**APPL061. Program Type — RPG. Process Phase — Extract.**

Used in conjunction with <PRECOL60> to combine record formats. Formats that start with the SLNO(*VAR) keyword are not combined and create their own record format. The last eight characters of the original record format name will be used as the name of

**MAKEPRECOL. Program Type — CLP. Process Phase — Combine.**

This proposal fills the file <PRECOLNW> with all possible record format combinations, which equals the result fileAWHOST. Combinations found in <PRECOLNW> will automatically be added to AWHOST. You may want to run this program once and then maintain <PRECOLNW> manually.

**PRECOLTOOL. Program Type — CLP. Process Phase — Before Collector Run.**

This proposal should be run before the Collector process is started. It provides a means to create your own combinations of record formats into pictures. The utility allows you to easily maintain the PRECOLNW file.

**APPL040. Program Type — RPG. Process Phase — Extract.**

Generates one record format for each possible variation in a psuedo-window of varying height. This enables each variation to be recognized as a popup window within the Picture Album. This proposal was created for a single customer and is not suitable for general use.

**APPL050. Program Type — CLP. Process Phase — Extract.**

This proposal is used to execute two proposals that must execute during the same Collector phase. In this example, the two proposals that are executed are <APPL031> and <APPL040>.

## Additional Proposals

The following proposals are other proposals you can use to influence the Collector run.

**APPL50**

Run multiple proposals in the same phase. Edit the APPL50 proposal to call the proposals you want to run. Then apply the proposal after the Extract phase. In this example of the APPL50 source, the APPL040 and APPL031 are specified to be run.

**APPL070**

Collect only specific display files.

**APPL080**

Omit certain formats from the collection. (From the standard collector options, you can only specify 16 formats.)

**EQUAL10**

Combine formats that have the same characters in the first part of the format name. (The default is the first 6 characters, but you can edit the program to change that number.) For example:

```
ITEM01
ITEM02
```

```
ITEM03
DEBTOR01
DEBTOR05
GENERAL01
ITEM04
ITEM01, 02, 03, and 04 are combined.
DEBTOR01 and 05 are combined.
```

**ERROR**

Force itself into an error after a certain phase so you can perform another task, such as looking at the intermediate files. You can then resume the collection by typing I (for ignore).

**Related topics**

Collector Proposal Descriptions

# Appendix C : Host Application DDS Keyword Listing

Below you will find an overview of keywords, that can be used in the DDS source for display files. Next to the keyword is stated ,the action taken by the Collector.

**ALARM**

No action taken. Passed.

**ALIAS**

No action taken

**ALTHELP**

The enabled key is passed.

**ALTNAME**

No action taken

**ALTPAGEDWN**

The enabled key is passed.

**ALTPAGEUP**

The enabled key is passed.

**ALWGPH**

No action taken.

**ALWROL**

No action taken.

**ASSUME**

No action taken.

**AUTO**

The appropriate attributes are passed.

**BLANKS**

No action taken.

**BLINK**

No action taken.

**BLKFOLD**

No action taken.

**CAxx**

Enabled function key is passed.

**CFxx**

Enabled function key is passed.

**CHANGE**

No action taken.

**CHCACCEL**

No action taken.

**CHCAVAIL**

No action taken. Also see SNGCHCFLD and MLTCHCFLD.

**CHCCTL**

No action taken. Also see SNGCHCFLD and MLTCHCFLD.

**CHCSLT**

No action taken. Also see SNGCHCFLD and MLTCHCFLD.

**CHCUNAVAIL**

No action taken. Also see SNGCHCFLD and MLTCHCFLD.

**CHGINPDFT or CHGINPDFT**

The appropriate attributes are passed.

**CHECK**

The appropriate attributes are passed.

**CHKMSGID**

See keyword ERRMSGID.

**CHOICE**

No action taken. Also see SNGCHCFLD and MLTCHCFLD.

**CHRID**

No action taken.

**CLEAR**

Enabled CLEAR key is passed.

**CLRL*ALL, CLRL*END, CLRL*NO, CLRLxx**

The record format containing this keyword will be handled according to the "CLRL" parameters set in the User Options.

**CMP**

No action taken.

**CNTFLD**

The field will be passed.

**COLOR**

The specified color is passed.

**COMP**

No action taken.

**CSRINPONLY**

No action taken.

**CSRLOC**

No action taken.

**DATE**

Date field is passed.

**DFT**

This may be the default value on an input field, which is passed. It may also be a constant.

**DFTVAL**

This is the default value on an output- or both-field, which is passed.

**DLTCHK**

No action taken.

**DLTEDT**

No action taken.

**DSPATR**

The attributes BL (blink), CS (column separator), HI (high intensity), ND (non display), PC (position cursor), PR (protect), RI (reversed image) and UL (underline) are passed. All other valid attributes are omitted.

**DSPMOD**

The appropriate screen sizes are passed.

**DSPRL**

No action taken.

**DSPSIZ**

The appropriate screen sized are passed.

**EDTCDE**

No action taken. The edit code used is found in the file field description of the object.

**DUP**

Enabled DUP key is passed.

**EDTMSK**

No action taken.

**EDTWRD**

No action taken. The edit word used is found in the file field description of the object.

**ENTFLDATR or ENTFLDATR**

The appropriate attributes are passed.

**ERASE**

No action taken.

**ERASEINP or ERASEINP**

No action taken.

**ERRMSG**

The message is passed.

**ERRMSGID**

The message is passed.

**ERRSFL**

This keyword may have it’s influence on the location of the messages.

**FLDCSRPRG**

No action taken.

**FLTFIXDEC**

No action taken.

**FLTPCN**

No action taken.

**FRCDTA**

No action taken.

**GETRETAIN**

No action taken.

**GRDATR**

No action taken.

**GRDBOX**

No action taken.

**GRDCLR**

No action taken.

**GRDLIN**

No action taken.

**GRDRCD**

No action taken.

**HELP**

Enabled HELP key is passed.

**HLPARA**

No action taken.

**HLPBDY**

No action taken.

**HLPCLR**

No action taken.

**HLPCMDKEY**

No action taken.

**HLPEXCLD**

No action taken.

**HLPDOC**

No action taken.

**HLPFULL**

No action taken.

**HLPID**

No action taken.

**HLPPNLGRP**

No action taken.

**HLPRCD**

No action taken.

**HLPRTN**

No action taken.

**HLPSCHIDX**

No action taken.

**HLPSEQ**

No action taken.

**HLPSHELF**

No action taken.

**HLPTITLE**

No action taken.

**HOME**

Enabled HOME key is passed.

**INDARA**

No action taken.

**INDTXT**

No action taken.

**INVITE**

No action taken.

**INZINP**

No action taken.

**INZRCD**

No action taken.

**KEEP**

No action taken.

**LOCK**

No action taken.

**LOGINP**

No action taken.

**LOGOUT**

No action taken.

**LOWER**

The LC attribute is passed.

**MDTOFF**

No action taken.

**MLTCHCFLD**

The field and literals will be passed simulating the enhanced display *NO option.

**MNUBAR**

No action taken.

**MNUBARSEP**

No action taken.

**MNUBARCHC**

No action taken.

**MNUBARDSP**

No action taken.

**MNUBARSW**

No action taken.

**MNUCNL**

No action taken.

**MOUBTN**

No action taken.

**MSGALARM**

No action taken.

**MSGCON**

The message is passed.

**MSGID**

The message is passed

**MSGLOC**

The row where error massages are displayed is passed.

**NOCCSID**

No action taken.

**OPENPRT**

No action taken.

**OVERLAY**

The record format is marked as "combine capable".

**OVRATR**

No action taken.

**OVRDTA**

No action taken.

**PAGEDOWN**

The enabled PAGEDOWN key is passed.

**PAGEUP**

The enabled PAGEUP key is passed.

**PASSCRD**

No action taken.

**PRINT**

The enabled PRINT key is passed.

**PROTECT**

No action is taken.

**PSHBTNCHC**

No action is taken.

**PSHBTNFLD**

No action is taken.

**PULLDOWN**

No action is taken.

**PUTOVR**

The record format is marked as "combine capable".

**PUTRETAIN**

No action taken.

**RANGE**

No action taken.

**RETCMDKEY**

No action taken.

**REF**

No action taken.

**REFFLD**

No action taken.

**RETKEY**

No action taken.

**RETCKSTS**

No action taken.

**ROLLDOWN**

The enabled ROLLDOWN key is passed.

**ROLLUP**

The enabled ROLLUP key is passed.

**RTNCSRLOC**

No action taken.

**RTNCSRLOC**

No action taken.

**RTNDTA**

No action taken.

**SETOF**

No action taken.

**SETOFF**

No action taken.

**SFL**

The record format defined by this keyword is always combined with its associated subfile control record format. The Collector will send all fields an/or texts in the subfile for each row of a subfile page.

**SFLCHCCTL**

No action taken.

**SFLCLR**

No action taken.

**SFLCSRPRG**

No action taken.

**SFLCSRRRN**

No action taken.

**SFLCTL**

See keyword SFL.

**SFLDLT**

No action taken.

**SFLDROP**

Two pictures will be generated, one folded and one unfolded.

**SFLDSP**

No action taken.

**SFLDSPCTL**

The appropriate "+", "MORE" and "BOTTOM" literals are passed.

**SFLEND**

The appropriate "+", "MORE" and "BOTTOM" literals are passed.

**SFLENTER**

No action taken.

**SFLFOLD**

Two pictures will be generated, one folded and one unfolded.

**SFLINZ**

No action taken.

**SFLLIN**

The number of blanks in between two records in the subfile is passed.

**SFLMLTCHC**

No action taken.

**SFLMODE**

No action taken.

**SFLMSG**

The message is passed.

**SFLMSGID**

The message is passed.

**SFLMSGKEY**

No action taken.

**SFLMSGRCD**

The message is passed.

**SFLNXTCHG**

No action taken.

**SFLPAG**

The number of subfile rows for a page is used. See keyword SFL.

**SFLPGMQ**

See keyword SFLMSGRCD().

**SFLRCDNBR**

No action taken.

**SFLRNA**

No action taken.

**SFLROLVAL**

No action taken.

**SFLRTNSEL**

No action taken.

**SFLCSROLL**

No action taken.

**SFLSIZ**

See keyword SFLPAG()

**SFLSNGCHC**

No action taken.

**SLNO*VAR**

The start row for this record format is unknown. Therefore the record format is marked as "not combine" capable.

**SLNOxx**

The start row for this record format is taken into account.

**SNGCHCFLD**

The field and literals will be passed simulating the enhanced display *NO option.

**TEXT**

No action taken.

**TIME**

The “time” field is passed.

**UNLOCK**

No action taken.

**USER**

The “user” field is passed.

**USRDFN**

No action taken.

**USRDSPMGT**

Unless stated otherwise the CLRL(*NO) is assumed for each record format.

**USRRSTDSP**

No action taken.

**VALNUM**

No action taken.

**VALUES**

No action taken.

**VLDCMDKEY**

No action taken.

**WDWBORDER**

The window borders (color, attributes and characters) are passed.

**WDWTITLE**

No action taken.

**WINDOW**

The window size is passed. The record format is marked as "not combine" capable.

**Related topics**

Collector DDS Keyword Handling

# Appendix D : Collector Work File Layouts

This chapter describes layouts of the Collector run working files. The work files hold the result each phase of the Collector process. Each file can be manipulated by user programs and proposals to affect the final outcome of the Collector run. The `AWHOST` stores the final results. A DDS description of all files used is present in the source file `QDDSSRC` in the library SEACOL.

**Related topics**

Collector Work Files

## AWOBJD - Object Description

```
5738SS1 V2R3M0  931217      Display File Field Description
 Input parameters
   File  . . . . . . . . . . . . . . . . . . . . :  AWOBJD
     Library . . . . . . . . . . . . . . . . . . :  SEACOL
 File Information
   File  . . . . . . . . . . . . . . . . . . . . :  AWOBJD
     Library . . . . . . . . . . . . . . . . . . :  SEACOL
   File location . . . . . . . . . . . . . . . . :  *LCL
   Externally described  . . . . . . . . . . . . :  Yes
   Number of record formats  . . . . . . . . . . :      1
   Type of file  . . . . . . . . . . . . . . . . :  Physical
   File creation date  . . . . . . . . . . . . . :  01-11-97
   Text 'description'. . . . . . . . . . . . . . :   Collector - objectdescriptions
 Record Format Information
   Record format . . . . . . . . . . . . . . . . :  QLIDOBJD
   Format level identifier . . . . . . . . . . . :  32E7BF3C24580
   Number of fields  . . . . . . . . . . . . . . :     85
   Record length . . . . . . . . . . . . . . . . :    519
   Format text . . . . . . . . . . . . . . . . . :  Object description
Field Level Information
Data        Field  Buffer    Buffer        Field    Column
Field       Type       Length  Length  Position        Usage    Heading

ODDCEN      CHAR            1       1         1        Both     Display century
Field text  . . . . . . . . . . . . . . . . . :  Display century
ODDDAT      CHAR            6       6         2        Both     Display date
Field text  . . . . . . . . . . . . . . . . . :  Display date: format- Job date format
ODDTIM      CHAR            6       6         8        Both     Display time
Field text  . . . . . . . . . . . . . . . . . :  Display time
ODLBNM      CHAR           10      10        14        Both     Library
Field text  . . . . . . . . . . . . . . . . . :  Library
ODOBNM      CHAR           10      10        24        Both     Object
Field text  . . . . . . . . . . . . . . . . . :  Object
ODOBTP      CHAR            8       8        34        Both     Object type
Field text  . . . . . . . . . . . . . . . . . :  Object type
ODOBAT      CHAR           10      10        42        Both     Object attribute
Field text  . . . . . . . . . . . . . . . . . :  Object attribute
ODOBFR      CHAR            1       1        52        Both     Storage freed
Field text  . . . . . . . . . . . . . . . . . :  Storage freed: 0-not freed,1-freed
ODOBSZ      PACKED      10  0       6        53        Both     Object size
Field text  . . . . . . . . . . . . . . . . . :  Object size
ODOBTX      CHAR           50      50        59        Both     Text description
Field text  . . . . . . . . . . . . . . . . . :  Text description
ODOBLK      CHAR            1       1       109        Both     Object locked
Field text  . . . . . . . . . . . . . . . . . :  Object locked: 0-not locked,1-locked
ODOBDM      CHAR            1       1       110        Both     Object damaged
Field text  . . . . . . . . . . . . . . . . . :  Object damaged: 0-not damaged,1-damaged
ODCCEN      CHAR            1       1       111        Both     Creation century
Field text  . . . . . . . . . . . . . . . . . :  Creation century
```

```
ODCDAT     CHAR            6       6       112        Both     Creation date
Field text  . . . . . . . . . . . . . . . . :  Creation date: format- mmddyy
ODCTIM     CHAR            6       6       118        Both     Creation time
Field text  . . . . . . . . . . . . . . . . :  Creation time
ODOBOW     CHAR           10      10       124        Both     Object owner
Field text  . . . . . . . . . . . . . . . . :  Object owner
ODSCEN     CHAR            1       1       134        Both     Save century
Field text  . . . . . . . . . . . . . . . . :  Save century
ODSDAT     CHAR            6       6       135        Both     Save date
Field text  . . . . . . . . . . . . . . . . :  Save date: format- mmddyy
ODSTIM     CHAR            6       6       141        Both     Save time
Field text  . . . . . . . . . . . . . . . . :  Save time
ODSCMD     CHAR           10      10       147        Both     Save command
Field text  . . . . . . . . . . . . . . . . :  Save command
ODSSZE     PACKED      10  0       6       157        Both     Saved size
Field text  . . . . . . . . . . . . . . . . :  Saved size
```

```
ODSSLT     ZONED        2  0       2       163        Both     Starting slot
Field text  . . . . . . . . . . . . . . . . :  Starting slot
ODSDEV     CHAR           10      10       165        Both     Save device
Field text  . . . . . . . . . . . . . . . . :  Save device
ODSV01     CHAR            6       6       175        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV02     CHAR            6       6       181        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV03     CHAR            6       6       187        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV04     CHAR            6       6       193        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV05     CHAR            6       6       199        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV06     CHAR            6       6       205        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV07     CHAR            6       6       211        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV08     CHAR            6       6       217        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV09     CHAR            6       6       223        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSV10     CHAR            6       6       229        Both     Saved volume
Field text  . . . . . . . . . . . . . . . . :  Saved volume
ODSVMR     CHAR            1       1       235        Both     More volumes
Field text  . . . . . . . . . . . . . . . . :  More volumes: 0-no more,
                                                1-more
ODRCEN     CHAR            1       1       236        Both     Restore century
Field text  . . . . . . . . . . . . . . . . :  Restore century
ODRDAT     CHAR            6       6       237        Both     Restore date
Field text  . . . . . . . . . . . . . . . . :  Restore date: format-mmddyy
ODRTIM     CHAR            6       6       243        Both     Restore time
Field text  . . . . . . . . . . . . . . . . :  Restore time
ODCPFL     CHAR            6       6       249        Both     System level
Field text  . . . . . . . . . . . . . . . . :  System level
ODSRCF     CHAR           10      10       255        Both     SRC file name
Field text  . . . . . . . . . . . . . . . . :  Source file name
ODSRCL     CHAR           10      10       265        Both     SRC file library
Field text  . . . . . . . . . . . . . . . . :  Source file library
ODSRCM     CHAR           10      10       275        Both     SRC file member
Field text  . . . . . . . . . . . . . . . . :  Source file member
ODSRCC     CHAR            1       1       285        Both     SRC change
                                                              century
Field text  . . . . . . . . . . . . . . . . :  Source change century
ODSRCD     CHAR            6       6       286        Both     SRC change date
Field text  . . . . . . . . . . . . . . . . :  Source change date:
                                                format-yymmdd
ODSRCT     CHAR            6       6       292        Both     SRC change time
```

```
Field text  . . . . . :  Source change time
ODCMNM     CHAR           7       7       298        Both     Compiler name
Field text  . . . . . :  Compiler name
ODCMVR     CHAR           6       6       305        Both     Compiler level
Field text  . . . . . :  Compiler level
ODOBLV     CHAR           8       8       311        Both     Object level
Field text  . . . . . :  Object level
ODUMOD     CHAR           1       1       319        Both     User modified
Field text  . . . . . : User modified:0-not modified,
                  1-modified
ODPPNM     CHAR           7       7       320        Both     LICPGM name
Field text  . . . . . :  LICPGM name
ODPPVR     CHAR           6       6       327        Both     LICPGM level
Field text  . . . . . :  LICPGM level
ODPCNR     CHAR           5       5       333        Both     PTF number
Field text  . . . . . :  PTF number
ODAPAR     CHAR           6       6       338        Both     APAR ID
Field text  . . . . . :  APAR ID
ODSSQN     ZONED        4  0       4       344        Both     Sequence number
Field text  . . . . . :  Sequence number
ODLCEN     CHAR           1       1       348        Both     Change century
Field text  . . . . . :  Change century
ODLDAT     CHAR           6       6       349        Both     Change date
Field text  . . . . . :  Change date: format- mmddyy
ODLTIM     CHAR           6       6       355        Both     Change time
Field text  . . . . . :  Change time
```

```
ODSFIL     CHAR          10      10       361        Both     Save file
Field text  . . . . . :  Save file
ODSFLB     CHAR          10      10       371        Both     Save file library
Field text  . . . . . :  Save file library
ODASP      ZONED        2  0       2       381        Both     ASP number
Field text  . . . . . :  ASP number
ODLBL      CHAR          17      17       383        Both     File label
Field text  . . . . . :  File label
ODPTFN     CHAR           7       7       400        Both     PTF ID
Field text  . . . . . :  PTF ID
ODOBSY     CHAR           8       8       407        Both     System name
Field text  . . . . . :  System name
ODCRTU     CHAR          10      10       415        Both     Created by user
Field text  . . . . . :  Created by user
ODCRTS     CHAR           8       8       425        Both     System created on
Field text  . . . . . :  System created on
ODUUPD     CHAR           1       1       433        Both     Usage updated
Field text  . . . . . :  Usage updated: Y-Yes, N-No
ODUCEN     CHAR           1       1       434        Both     Last Used century
Field text  . . . . . :  Last used century
ODUDAT     CHAR           6       6       435        Both     Last Used date
Field text  . . . . . :  Last used date: format- mmddyy
ODUCNT     PACKED       5  0       3       441        Both     Days Used count
Field text  . . . . . :  Days used count
ODTCEN     CHAR           1       1       444        Both     Reset century
Field text  . . . . . :  Reset century
ODTDAT     CHAR           6       6       445        Both     Reset date
Field text  . . . . . :  Reset date: format- mmddyy
ODODMN     CHAR           2       2       451        Both     Object domain
Field text  . . . . . :  Object domain: *S - System, *U - User
ODCPVR     CHAR           6       6       453        Both     System version
Field text  . . . . . :  System version
ODCVRM     CHAR           6       6       459        Both     Compiler version
Field text  . . . . . :  Compiler version
ODPVRM     CHAR           6       6       465        Both     LICPGM version
Field text  . . . . . :  LICPGM version
ODCPRS     CHAR           1       1       471        Both     Compression status
Field text  . . . . . :  Compression status
```

```
ODOASP     CHAR            1       1       472        Both     Overflowed ASP
Field text  . . . . . . . . . . . . . . . . :  Overflowed ASP: 0-No,1-Yes
ODAAPI     CHAR            1       1       473        Both     Allow API change
Field text  . . . . . . . . . . . . . . . . :  Allow change by API: 0-No,1-Yes
ODAPIC     CHAR            1       1       474        Both     Changed by API
Field text  . . . . . . . . . . . . . . . . :  Changed by API: 0-not changed,1-changed
ODUATR     CHAR           10      10       475        Both     User-defined attribute
Field text  . . . . . . . . . . . . . . . . :  User-defined attribute
ODACEN     CHAR            1       1       485        Both     Save active century
Field text  . . . . . . . . . . . . . . . . :  Save active century
ODADAT     CHAR            6       6       486        Both     Save active date
Field text  . . . . . . . . . . . . . . . . :  Save active date: format- mmddyy
ODATIM     CHAR            6       6       492        Both     Save active time
Field text  . . . . . . . . . . . . . . . . :  Save active time
ODAUDT     CHAR           10      10       498        Both     Auditing value
Field text  . . . . . . . . . . . . . . . . :  Object auditing value
ODSIZU     PACKED      10  0       6       508        Both     Object size in units
Field text  . . . . . . . . . . . . . . . . :  Object size in units
ODBPUN     PACKED      10  0       6       514        Both     Bytes per unit
Field text  . . . . . . . . . . . . . . . . :  Bytes per unit
```

# AWMB - Member Listing

```
5738SS1 V2R3M0  931217      Display File Field Description
 Input parameters
   File  . . . . . . . . . . . . . . . . . . :  AWMB
     Library . . . . . . . . . . . . . . . . :  SEACOL
 File Information
   File  . . . . . . . . . . . . . . . . . . :  AWMB
     Library . . . . . . . . . . . . . . . . :  SEACOL
   File location . . . . . . . . . . . . . . :  *LCL
   Externally described  . . . . . . . . . . :  Yes
   Number of record formats  . . . . . . . . :      1
   Type of file  . . . . . . . . . . . . . . :  Physical
   File creation date  . . . . . . . . . . . :  01-11-97
   Text 'description'. . . . . . . . . . . . :   Collector - memberlist to treat
 Record Format Information
   Record format . . . . . . . . . . . . . . :  QWHFDML
   Format level identifier . . . . . . . . . :  51852FE30C648
   Number of fields  . . . . . . . . . . . . :     31
   Record length . . . . . . . . . . . . . . :    201
   Format text . . . . . . . . . . . . . . . :  DSPFD format for TYPE *MBRLIST
Field Level Information
Data        Field  Buffer    Buffer        Field    Column
Field       Type       Length  Length  Position        Usage    Heading
MLRCEN     CHAR            1       1         1        Both     Retrieval century
Field text  . . . . . . . . . . . . . . . . :  Retrieval century:  0=20th, 1=21st
MLRDAT     CHAR            6       6         2        Both     Retrieval date
Field text  . . . . . . . . . . . . . . . . :  Retrieval date: year/month/day
MLRTIM     CHAR            6       6         8        Both     Retrieval time
Field text  . . . . . . . . . . . . . . . . :  Retrieval time: hour/minute/second
MLFILE     CHAR           10      10        14        Both     File
Field text  . . . . . . . . . . . . . . . . :  File
MLLIB      CHAR           10      10        24        Both     Library
Field text  . . . . . . . . . . . . . . . . :  Library
MLFTYP     CHAR            1       1        34        Both     Type of file
Field text  . . . . . . . . . . . . . . . . :  P=PF, L=LF, R=DDM PF, S=DDM LF
MLFILA     CHAR            4       4        35        Both     File attribute
Field text  . . . . . . . . . . . . . . . . :  File attribute: *PHY or *LGL
MLMXD      CHAR            3       3        39        Both     Reserved
Field text  . . . . . . . . . . . . . . . . :  Reserved
MLFATR     CHAR            6       6        42        Both     File attribute
Field text  . . . . . . . . . . . . . . . . :  File attribute:  PF, LF, PF38,or LF38
```

```
MLSYSN     CHAR            8       8        48        Both     System name
Field text  . . . . . . . . . . . . . . . . : System Name (Source System,if file DDM)
MLASP      PACKED       3  0       2        56        Both     ASP auxiliary
                                                                storage pool ID
Field text  . . . . . . . . . . . . . . . . :  Auxiliary storage pool ID:1=System ASP
MLRES      CHAR            4       4        58        Both     Reserved
Field text  . . . . . . . . . . . . . . . . :  Reserved
MLNOMB     PACKED       5  0       3        62         Both     Number of members
Field text  . . . . . . . . . . . . . . . . :  Number of members
MLNAME     CHAR           10      10        65        Both     Member
Field text  . . . . . . . . . . . . . . . . :  Member
MLNRCD     PACKED      10  0       6        75        Both     Number of records
Field text  . . . . . . . . . . . . . . . . :  Current number of records
MLNDTR     PACKED      10  0       6        81        Both     Deleted records
Field text  . . . . . . . . . . . . . . . . :  Number of deleted records
MLSIZE     PACKED      10  0       6        87        Both     Data Space size
Field text  . . . . . . . . . . . . . . . . :  Data space and index size in bytes,
                                               -1 = See MLSIZ
MLSEU      CHAR            4       4        93        Both     Source type
Field text  . . . . . . . . . . . . . . . . :  Source type for S/38 View as it
                                               appeared on S/38
MLCCEN     CHAR            1       1        97        Both    Member creation century
Field text  . . . . . . . . . . . . . . . . :  Member creation century: 0=20th,1=21st
MLCDAT     CHAR            6       6        98        Both     Member creation
```

```
Field text  . . . . . . . . . . . . . . . . : Member creation century:year/month/
                             day
MLCHGC     CHAR            1       1       104        Both     Last change
                           century
Field text  . . . . . . . . . . . . . . . . :  Last change century: 0=20th,
                           1=21st
MLCHGD     CHAR            6       6       105        Both     Last change date
Field text  . . . . . . . . . . . . . . . . :  Last change date: year/month/day
MLCHGT     CHAR            6       6       111        Both     Last change time
Field text  . . . . . . . . . . . . . . . . :  Last change time: hour/minute/
                           second
MLMTXT     CHAR           50      50       117        Both     TEXT
                          'description'
Field text  . . . . . . . . . . . . . . . . :  Text 'description'
MLSEU2     CHAR           10      10       167        Both     Source type
Field text  . . . . . . . . . . . . . . . . :  Source type
MLUCEN     CHAR            1       1       177        Both     Last Used century
Field text  . . . . . . . . . . . . . . . . :  Last Used Century: 0=20th, 1=21st
MLUDAT     CHAR            6       6       178        Both     Last Used date
Field text  . . . . . . . . . . . . . . . . :  Last Used Date: year/month/day
MLUCNT     PACKED       5  0       3       184        Both     Days Used Count
Field text  . . . . . . . . . . . . . . . . :  Days Used Count
MLTCEN     CHAR            1       1       187        Both    Usage Data reset
                          century
Field text  . . . . . . . . . . . . . . . . :  Usage Data Reset Century:0=20th,
                          1=21st
MLTDAT     CHAR            6       6       188        Both     Usage Data reset
                          date
Field text  . . . . . . . . . . . . . . . . :  Usage Data Reset Date: year/month/
                           day
MLSIZ2     PACKED      15  0       8       194        Both     Data Space size
Field text  . . . . . . . . . . . . . . . . :  Data space and index size in bytes
```

# AWFTM - Format Attributes

```
5738SS1 V2R3M0  931217      Display File Field Description
 Input parameters
```

```
   File  . . . . . . . . . . . . . . . . . . . . :  AWFMT
 File Information
   File  . . . . . . . . . . . . . . . . . . . . :  AWFMT
   File location . . . . . . . . . . . . . . . . :  *LCL
   Externally described  . . . . . . . . . . . . :  Yes
   Number of record formats  . . . . . . . . . . :      1
   Type of file  . . . . . . . . . . . . . . . . :  Physical
   File creation date  . . . . . . . . . . . . . :  01-11-97
   Text 'description'. . . . . . . . . . . . . . :   Collector - format attributes
Record Format Information
   Record format . . . . . . . . . . . . . . . . :  AWFREC
   Format level identifier . . . . . . . . . . . :  426D5417B691F
   Number of fields  . . . . . . . . . . . . . . :     47
   Record length . . . . . . . . . . . . . . . . :    225
 Field Level Information
AWFFIL     CHAR           10      10         1        Both     FILE
Field text  . . . . . . . . . . . . . . . . . . :  Object file name
AWFLIB     CHAR           10      10        11        Both     LIBRAR
Field text  . . . . . . . . . . . . . . . . . . :  Library name
AWFFRM     CHAR           10      10        21        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . . . :  Record format name
AWFSIZ     ZONED        3  0       3        31        Both     DSPWID
Field text  . . . . . . . . . . . . . . . . . . :  Page size width of record format
AWFHGT     ZONED        3  0       3        34        Both     DSPHGT
Field text  . . . . . . . . . . . . . . . . . . :  Page size height of record format
AWFGRP     PACKED       5  0       3        37        Both     GRPHCS
Field text  . . . . . . . . . . . . . . . . . . :  Graphic character set
AWFCOD     PACKED       5  0       3        40        Both     CODPAG
Field text  . . . . . . . . . . . . . . . . . . :  Code page
AWFFNC     CHAR           32      32        43        Both     ENABLE
Field text  . . . . . . . . . . . . . . . . . . :  Enabled function keys for record format
AWFSLN     ZONED        3  0       3        75        Both     STRLIN
Field text  . . . . . . . . . . . . . . . . . . :  Start row record format
AWFELN     ZONED        3  0       3        78        Both     ENDLIN
Field text  . . . . . . . . . . . . . . . . . . :  Last row record format
AWFOVL     ZONED        1  0       1        81        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . . . :  Overlay code record format 0/1=N/Y
AWFINC     ZONED        1  0       1        82        Both     AWFINC
Field text  . . . . . . . . . . . . . . . . . . :  Include code for combine
0 = record format contains WINDOW keyword
      1 = record format contains SLNO(*VAR) keyword
      2 = record format contains CLRL(*NO) keyword
      3 = record format contains CLRL(xx) keyword, not to bottom row
      4 = record format contains CLRL(*NO) keyword, only bottom row
      5 = record format contains CRLR(xx) keyword, up to bottom row
      6 = record format contains CLRL(xx) keyword, from start up to bottom row
      7 = record format contains keyword CLRL(*ALL) keyword
      8 =    record format does not contain any keyword mentioned above
AWFCLN     ZONED        3  0       3        83        Both     LINE
Field text  . . . . . . . . . . . . . . . . . . :  Number of rows to clear
AWFWDW     ZONED        1  0       1        86        Both     AWWDW
Field text  . . . . . . . . . . . . . . . . . . :  Record format is window 0/1 = N/Y
AWFMSL     ZONED        3  0       3        87        Both     LINE
Field text  . . . . . . . . . . . . . . . . . . :  Row for message location
```

```
AWFFM1     CHAR           10      10        90        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . . . :  Record format name 1
AWFSL1     ZONED        3  0       3       100        Both     LINE
Field text  . . . . . . . . . . . . . . . . . . :  Start row record format 1
AWFEL1     ZONED        3  0       3       103        Both     LINE
Field text  . . . . . . . . . . . . . . . . . . :  Last row record format 1
AWFOV1     ZONED        1  0       1       106        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . . . :  Overlay code record format 1
AWFFM2     CHAR           10      10       107        Both     FRMNAM
```

```
Field text  . . . . . . . . . . . . . . . . :  Record format name 2
AWFSL2     ZONED        3  0       3       117        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Start row record format 2
AWFEL2     ZONED        3  0       3       120        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Last row record format 2
AWFOV2     ZONED        1  0       1       123        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . :  Overlay code record format 2
AWFFM3     CHAR          10      10       124        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . :  Record format name 3
AWFSL3     ZONED        3  0       3       134        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Start row record format 3
AWFEL3     ZONED        3  0       3       137        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Last row record format 3
AWFOV3     ZONED        1  0       1       140        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . :  Overlay code record format 3
AWFFM4     CHAR          10      10       141        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . :  Record format name 4
AWFSL4     ZONED        3  0       3       151        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Start row record format 4
AWFEL4     ZONED        3  0       3       154        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Last row record format 4
AWFOV4     ZONED        1  0       1       157        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . :  Overlay code record format 4
AWFFM5     CHAR          10      10       158        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . :  Record format name 5
AWFSL5     ZONED        3  0       3       168        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Start row record format 5
AWFEL5     ZONED        3  0       3       171        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Last row record format 5
AWFOV5     ZONED        1  0       1       174        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . :  Overlay code record format 5
AWFFM6     CHAR          10      10       175        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . :  Record format name 6
AWFSL6     ZONED        3  0       3       185        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Start row record format 6
AWFEL6     ZONED        3  0       3       188        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Last row record format 6
AWFOV6     ZONED        1  0       1       191        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . :  Overlay code record format 6
AWFFM7     CHAR          10      10       192        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . :  Record format name 7
AWFSL7     ZONED        3  0       3       202        Both     LINEField text
Field text. . . . . . . . . . . . . . . . . :  Start row record format 7
AWFEL7     ZONED        3  0       3       205        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Last row record format 7
AWFOV7     ZONED        1  0       1       208        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . :  Overlay code record format 7
AWFFM8     CHAR          10      10       209        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . :  Record format name 8
AWFSL8     ZONED        3  0       3       219        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Start row record format 8
AWFEL8     ZONED        3  0       3       222        Both     LINE
Field text  . . . . . . . . . . . . . . . . :  Last row record format 8
AWFOV8     ZONED        1  0       1       225        Both     AWOVRL
Field text  . . . . . . . . . . . . . . . . :  Overlay code record format 8
```

# AWTXT - Single DDS Texts

```
5738SS1 V3R310  951115      Display File Field Description
 Input parameters
   File  . . . . . . . . . . . . . . . . . . . :  AWTXT
     Library . . . . . . . . . . . . . . . . . :  SEACOL
 File Information
```

```
   File  . . . . . . . . . . . . . . . . . . . . :  AWTXT
     Library . . . . . . . . . . . . . . . . . . :  SEACOL
   File location . . . . . . . . . . . . . . . . :  *LCL
   Externally described  . . . . . . . . . . . . :  Yes
   Number of record formats  . . . . . . . . . . :      1
   Type of file  . . . . . . . . . . . . . . . . :  Physical
   File creation date  . . . . . . . . . . . . . :  01-11-97
   Text 'description'. . . . . . . . . . . . . . :   Collector - single DDS-texts to PC
 Record Format Information
   Record format . . . . . . . . . . . . . . . . :  AWTREC
   Format level identifier . . . . . . . . . . . :  2BE3F0483EC39
   Number of fields  . . . . . . . . . . . . . . :     22
   Record length . . . . . . . . . . . . . . . . :    442
Field Level Information
AWLIBR     CHAR          10      10         1        Both     LIBRAR
Field text  . . . . . . . . . . . . . . . . . :  Library name
AWFILE     CHAR          10      10        11        Both     FILE
Field text  . . . . . . . . . . . . . . . . . :  Source file name
AWMEMB     CHAR          10      10        21        Both     MEMBER
Field text  . . . . . . . . . . . . . . . . . :  Source member name
AWSTNR     ZONED        6  0       6        31        Both     STMCNT
Field text  . . . . . . . . . . . . . . . . . :  Statement counter
AWFORM     CHAR          10      10        37        Both     FRMNAM
Field text  . . . . . . . . . . . . . . . . . :  Record format name
AWLINE     ZONED        3  0       3        47        Both     LINE
Field text  . . . . . . . . . . . . . . . . . :  Row for this field or text
AWCOLM     ZONED        3  0       3        50        Both     COLUMN
Field text  . . . . . . . . . . . . . . . . . :  Column for this field or text
AWFLDN     CHAR          10      10        53        Both     FIELD
Field text  . . . . . . . . . . . . . . . . . :  Field name
AWFLDT     CHAR           1       1        63        Both     FLDTYP
Field text  . . . . . . . . . . . . . . . . . :  Field type B/A/S/P/F/O/J/E/H
AWKIND     ZONED        2  0       2        64        Both     AWHKND
Field text  . . . . . . . . . . . . . . . . . :  Kind of field or text
AWDLNG     ZONED        5  0       5        66        Both     TXTLE5
Field text  . . . . . . . . . . . . . . . . . :  Edited length of field or text
AWLNGT     ZONED        5  0       5        71        Both     TXTLE5
Field text  . . . . . . . . . . . . . . . . . :  Length for field or text
AWNDEC     ZONED        2  0       2        76        Both     NRDEC
Field text  . . . . . . . . . . . . . . . . . :  Number of decimals this field
AWVARL     ZONED        1  0       1        78        Both     AWHVAR
Field text  . . . . . . . . . . . . . . . . . :  Variable length 0/1=N/Y
AWEXT      CHAR          80      80        79        Both     INDEXT
Field text  . . . . . . . . . . . . . . . . . :  Extra indicators for field or text
AWIND      CHAR          10      10       159        Both     INDUSE
Field text  . . . . . . . . . . . . . . . . . :  Last indicators for field or text
AWATTR     CHAR          20      20       169        Both     ATTRIB
Field text  . . . . . . . . . . . . . . . . . :  Attributes for field or text
AWFLIO     CHAR           1       1       189        Both     IOATTR
Field text  . . . . . . . . . . . . . . . . . :  IO-attribute I/O/B/N
AWMSGF     ZONED        1  0       1       190        Both     MSGFOU
Field text  . . . . . . . . . . . . . . . . . :  Message function found 0/1=N/Y
AWLAST     ZONED        1  0       1       191        Both     AWLAST
Field text  . . . . . . . . . . . . . . . . . :  Sort field for <AWHOST>0/1=N/Y
AWMESS     CHAR         250     250       192        Both     MESSAGE
Field text  . . . . . . . . . . . . . . . . . :  Literal text
AWSSIZ     ZONED        1  0       1       442        Both     AWSIZ
Field text  . . . . . . . . . . . . . . . . . :  0 is primary, 1 is secondary size
```

# AWHOST — Layout

```
                                                                               
================================================================================
```

```
         S E A G U L L    B U S I N E S S    S O F T W A R E
                     C O L L E C T O R

         F I L E   L A Y - O U T   V E R S I O N    0 0 0 0 0 0 0 5
         R E C O R D   L E N G T H   =   3 1 4   B Y T E S
=========================================================================
         H O S T - F I L E   T O   S E N D   T O   P C
=========================================================================
                                    REF(SEACOL/AWREF)
       R AWHREC
         HSTFLD    R               REFFLD(HSTFLD)
DISPL  R AWHR00
 001     AWHT00    R               REFFLD(RECTYP)
                                   <00> = file header
 003     AWNX00    R               REFFLD(RECTYP)
                                   Record type of continuing record
                                   <00> = no continuing record
 005     AWHID0    R               REFFLD(LAYOUT)
                                   Layout version <AWHOST> (00000005)
 013     AWHDAT    R               REFFLD(CRTCDA)
                                   Creation date C-YY-MM-DD
 020     AWHTIM    R               REFFLD(CRTTIM)
                                   Creation time HH-MM-SS
 026     FILL01        16A         TEXT('FILLER')
 042     AWSRLN    R               REFFLD(AWSRLN)
                                   Serial number of AS/400 creating
 050     AWCOMP        30A         TEXT('COMPANY NAME')
           Company name (for future use)
 080     AWHNR0    R               REFFLD(AWHREC)
                                   Total number of records type 00
 090     AWHNR1    R               REFFLD(AWHREC)
                                   Total number of records type 01
 100     AWHNR2    R               REFFLD(AWHREC)
                                   Total number of records type 02
 110     AWHNR3    R               REFFLD(AWHREC)
                                   Total number of records type 03
 120     AWHNR4    R               REFFLD(AWHREC)
                                   Total number of records type 04
 130     AWHNR5    R               REFFLD(AWHREC)
                                   Total number of records type 05
 140     AWHNR6    R               REFFLD(AWHREC)
                                   Total number of records type 06
 150     AWHNR7    R               REFFLD(AWHREC)
                                   Total number of records type 07
 160     AWHNR8    R               REFFLD(AWHREC)
                                   Total number of records type 08
 170     AWHNR9    R               REFFLD(AWHREC)
                                   Total number of records type 09
 180     AWHVER    R               REFFLD(AWVER)
                                    Collector version
 194     DBMARK    R               REFFLD(DBMARK)
                                   DBCS-MARK (FOR INTERNAL USE ONLY)
 200     FILL02       115A         TEXT('FILLER')
=========================================================================
DISPL  R AWHR01
         AWHT01    R               REFFLD(RECTYP)
                                   <01> = Reserved for future use
 003     FILL11       250A         TEXT('FILLER')
 253     FILL12        62A         TEXT('FILLER')
=========================================================================
```

```
DISPL  R AWHR02
 001     AWHT02    R               REFFLD(RECTYP)
                                   <02> = reserved for future use
```

```
 003      FILL21       250A         TEXT('FILLER')
 253      FILL22        62A         TEXT('FILLER')
DISPL R AWHRO3
 001      AWHT03    R               REFFLD(RECTYP)
                                    <03> = picture header
 003      AWNX03    R               REFFLD(RECTYP)
                                    Record type of continuing record
                                    <00> = no continuing record
 005      AWHID3    R               REFFLD(AWHNAM)
                                    Internal name of picture
 013      AWHLIB    R               REFFLD(LIBRAR)
                                    Library of picture
 023      AWHMBR    R               REFFLD(MEMBER)
                                    Member name of picture
 033      AWHKEY    R               REFFLD(ENABLE)
              <0> = not active
                                    <1> = active
                                    <2> = conditional active
                                    byte 01 - 24 = function key 01 - 24
                                    byte 25      = (PRINT) key
                                    byte 26      = (HOME) key
                                    byte 27      = (ROLLUP) key
                                    byte 28      = (ROLLDOWN) key
                                    byte 29      = (CLEAR) key
                                    byte 30      = (HELP) key
                                    byte 31      = (DUP) key
                                    byte 32      = reserved for future use
                                    Number of records type 06 for this picture
 077      AWGRPH    R               REFFLD(AWGRPH)
                                    Graphical character set this picture
 082      AWCODP    R               REFFLD(AWCODP)
                                    Code page this picture
 087      AWFMT1    R               REFFLD(FRMNAM)
                                    Record format name 1
 097      AWSLN1    R               REFFLD(LINE)
                                    Start row record format 1
 100      AWELN1    R               REFFLD(LINE)
                                    Last row record format 1
 103      AWOVL1    R               REFFLD(AWOVRL)
                                    Overlay code record format 1
                                   <0> = not, <1> = yes
 065      AWHSIZ    R               REFFLD(DSPWID)
                                    Width of picture
 068      AWHHGT    R               REFFLD(DSPHGT)
                                    Heigth of picture
 071      AWHNRT    R               REFFLD(AWRSCR)
 104      AWFMT2    R               REFFLD(FRMNAM)
                                    Record format name 2
 114      AWSLN2    R               REFFLD(LINE)
                                                    Start row record format 2
 117      AWELN2    R               REFFLD(LINE)
                                    Last row record format 2
 120      AWOVL2    R               REFFLD(AWOVRL)
                                    Overlay code record format 2
                                    <0> = not, <1> = yes
 121      AWFMT3    R               REFFLD(FRMNAM)
                                    Record format name 3
 131      AWSLN3    R               REFFLD(LINE)
                                    Start row record format 3
 134      AWELN3    R               REFFLD(LINE)
                                    Last row record format 3
 137      AWOVL3    R               REFFLD(AWOVRL)
                                    Overlay code record format 3
```

```
 138      AWFMT4    R               REFFLD(FRMNAM)
```

```
                                    Record format name 4
148      AWSLN4    R                REFFLD(LINE)
                                    Start row record format 4
151      AWELN4    R                REFFLD(LINE)
                                    Last row record format 4
154      AWOVL4    R                REFFLD(AWOVRL)
                                    Overlay code record format 4
                                    <0> = not, <1> = yes
155      AWFMT5    R                REFFLD(FRMNAM)
                                    Record format name 5
165      AWSLN5    R                REFFLD(LINE)
                                    Start row record format 5
168      AWELN5    R                REFFLD(LINE)
                                    Last row record format 5
171      AWOVL5    R                REFFLD(AWOVRL)
                                    Overlay code record format 5
                                    <0> = not, <1> = yes
172      AWFMT6    R                REFFLD(FRMNAM)
                                    Record format name 6
182      AWSLN6    R                REFFLD(LINE)
              Start row record format 6
185      AWELN6    R                REFFLD(LINE)
                                    Last row record format 6
188      AWOVL6    R                REFFLD(AWOVRL)
                                    Overlay code record format 6
                                    <0> = not, <1> = yes
189      AWFMT7    R                REFFLD(FRMNAM)
                                    Record format name 7
199      AWSLN7    R                REFFLD(LINE)
                                    Start row record format 7
202      AWELN7    R                REFFLD(LINE)
                                    Last row record format 7
205      AWOVL7    R                REFFLD(AWOVRL)
                                    Overlay code record format 7
                                    <0> = not, <1> = yes
206      AWFMT8    R                REFFLD(FRMNAM)
                                    Record format name 8
216      AWSLN8    R                REFFLD(LINE)
                                    Start row record format 8
219      AWELN8    R                REFFLD(LINE)
                                    Last row record format 8
222      AWOVL8    R                REFFLD(AWOVRL)
                                    Overlay code record format 8
                                    <0> = not, <1> = yes
223      AWWDW     R                REFFLD(AWWDW)
                                    Record format is window
                                    <0> = not
                                    <1> = yes, fixed position
                                    <2> = yes, variable position
224      AWWUPP    R                REFFLD(BORDER)
                                    Top border of window
                                                      <0> = variable top border
227      AWWLFT    R                REFFLD(BORDER)
                                    Left border of window
                                    <0> = variable left window
230      AWWLOW    R                REFFLD(BORDER)
                                    Bottom border of window
233      AWWRGT    R                REFFLD(BORDER)
                                    Right border of window
236      AWWSIZ    R                REFFLD(DSPWID)
                                    Width of window including borders
                                    equals field <AWHSIZ> if no window
239      AWWHGT    R                REFFLD(DSPHGT)
                                    Heigth of window including borders
                                    equals field <AWHHGT> if no window
242      AWWCH3    R                REFFLD(AWWCHR) Border characters of window
```

```
 250      AWWAT3    R               REFFLD(ATTRIB)
                                    Attributes of border characters
                <0> = not active
                                     <1> = active
                                     <2> = conditional active
                                     byte 01 = color <BLACK>
                                     byte 02 = color <BLUE>
                                     byte 03 = color <BROWN>
                                     byte 04 = color <GREEN>
                                     byte 05 = color <PINK>
                                     byte 06 = color <RED>
                                     byte 07 = color <TURQUOISE>
                                     byte 08 = color <WHITE>
               byte 09 = color <YELLOW>
                                     byte 10 = attribute <BLINKING>
                                     byte 11 = attribute <COLUMN SEPARATOR>
                                     byte 12 = attribute <HIGH INTENSITY>
                                     byte 13 = attribute <NON DISPLAY>
                                     byte 14 = attribute <POSITION CURSOR>
                                     byte 15 = attribute <REVERSED IMAGE>
                                     byte 16 = attribute <UNDERLINE>
                                     byte 17 = attribute <PROTECT>
                                     byte 18 = reserved for future use
                                     byte 19 = reserved for future use
                                     byte 20 = reserved for future use
 270      AWBUPP    R               REFFLD(BORDER)
                                    Heigth of top border of window
                                    <1> = fixed value
 273      AWBLFT    R               REFFLD(BORDER)
                                    Width of left border of window
                                    <3> = fixed value
 276      AWBLOW    R               REFFLD(BORDER)
                                    Heigth of bottom border of window
                                    <1> = fixed value
 279      AWBRGT    R               REFFLD(BORDER)
                                    Width of right border of window
                                    <3> = fixed value
 282      AWMSGL    R               REFFLD(LINE)
                                     MESSAGE LOCATION
 285      AWHDRL    R               REFFLD(RITOLE)
                                     FORMAT DISPLAYED RIGHT TO LEFT
                                     <0> = NOT, <1> = YES
 286      AWHCRL    R               REFFLD(RITOLE)
                                     CURSOR MOVES RIGHT TO LEFT IN FORMAT
                                     <0> = NOT, <1> = YES
 287      AWWDW2   R               REFFLD(AWWDW)
                                     WINDOWFORMAT
                                     COPY FROM FIELD <AWWDW> AT BYTE 223
                                     <0> = NOT
                                     <1> = YES, FIXED POSITION
                                     <2> = YES, VARIABLE POSITION
                                     <3> = YES, VARIABLE POSITION
                                           INCLUDING WINDOW-TITLE
 288      FILL3         27A         TEXT('FILLER')
*****************************************************************************
DISPL    R AWHRO4
 001      AWHT04    R               REFFLD(RECTYP)
                                    <04> = border attributes for window
 003      AWNX04    R               REFFLD(RECTYP)
                                    Record type of continuing record
                                    <00> = no continuing record
 005      AWWCH4    R               REFFLD(AWWCHR)
                                    Border characters of window
```

```
 013      AWWAT4    R               REFFLD(ATTRIB)
               Attributes of border characters
                                    <0> = not active
                                    <1> = active
                                    <2> = conditional active
                                    byte 01 = color <BLACK>
                                    byte 02 = color <BLUE>
                                    byte 03 = color <BROWN>
                                    byte 04 = color <GREEN>
                                    byte 05 = color <PINK>
                                    byte 06 = color <RED>
                                    byte 07 = color <TURQUOISE>
                                    byte 08 = color <WHITE>
                                    byte 09 = color <YELLOW>
                                    byte 10 = attribute <BLINKING>
                                    byte 11 = attribute <COLUMN SEPARATOR>
                                    byte 12 = attribute <HIGH INTENSITY>
                                    byte 13 = attribute <NON DISPLAY>
                                    byte 14 = attribute <POSITION CURSOR>
                                    byte 15 = attribute <REVERSED IMAGE>
                                    byte 16 = attribute <UNDERLINE>
                                    byte 17 = attribute <PROTECT>
                                    byte 18 = reserved for future use
                                    byte 19 = reserved for future use
                                     byte 20 = reserved for future use
 033      FILL41        32A         TEXT('FILLER')
 065      FILL42       250A         TEXT('FILLER')
=========================================================================
DISPL   R AWHR05
 001      AWHT05    R               REFFLD(RECTYP)
                                    <05> = reserved for future use
 003      FILL51       250A         TEXT('FILLER')
 253      FILL52        62A         TEXT('FILLER')
=========================================================================
DISPL   R AWHR06
 001      AWHT06    R               REFFLD(RECTYP)
                                    <06> = field- or text-definitions
 003      AWNX06    R               REFFLD(RECTYP)
                                    record type of continuing record
                                    <00> = no continuing record
 005      AWHKND    R               REFFLD(AWHKND)
DISPL   R AWHR06
 001      AWHT06    R               REFFLD(RECTYP)
                                    <06> = field- or text-definitions
 003      AWNX06    R               REFFLD(RECTYP)
                                    record type of continuing record
                                    <00> = no continuing record
 005      AWHKND    R               REFFLD(AWHKND)
               <00> = field not in subfile
                                    <01> = field in subfile header
                                    <02> = field in subfile
                                    <10> = fixed text not in subfile
                                    <11> = fixed text in subfile header
                                    <12> = fixed text in subfile
                                    <20> = text using <MSGCON>-keyword
                                           not in subfile
                                    <21> = text using <MSGCON>-keyword
                                           in subfile header
                                    <22> = text using <MSGCON>-keyword
                                           in subfile
                                    <30> = text using <MSGID>-keyword
                                           not in subfile
                                    <31> = text using <MSGID>-keyword
                                           in subfile header
                                    <32> = text using <MSGID>-keyword
                                           in subfile <impossible>
```

```
                                    <40> = text using <ERRMSG>-keyword
                                    <50> = text using <ERRMSGID>-keyword
                                    <60> = text using <SFLMSG>-keyword
                                    <70> = text using <SFLMSGID>-keyword
                                    <80> = subfile end indication <+>
                                    <81> = subfile end indication <MORE>
                                    <82> = subfile end indication <BOTTOM>
```

```
 007      AWHCO6    R               REFFLD(AWHCON)
                                    Conditional display
                                    <0> = not, <1> = yes
 008      AWHIN6    R               REFFLD(INDUSE)
                                    byte 01      = more indicators active
                                    byte 02 - 10 = active indicators
 018      AWHFTP    R               REFFLD(FLDTYP)
             < > = text
                                    <A> = character data
                                    <B> = binary data
                                    <E> = DBCS-either data
                                    <F> = floating point data
                                    <G> = DBCS-graphic data
                                    <H> = hexadecimal data
                                    <J> = DBCS-only data
                                    <L> = date data
                                    <O> = DBCS-open data
                                    <P> = packed decimal data
                                    <S> = zoned decimal data
                                    <T> = time data
                                    <Z> = time stamp
019      AWHFIO    R               REFFLD(IOATTR)
                                    <B> = input and output allowed
                                    <I> = only input allowed
                                    <N> = no input or output allowed
                                    <O> = only output allowed
020      AWHATR    R               REFFLD(ATTRIB)
                                    <0> = not active
                                    <1> = active
                                    <2> = conditional active
                                    byte 01 = Right to left input
                                    byte 02 = color <BLUE>
                                    byte 03 = DUP key allowed
                                    byte 04 = color <GREEN>
                                    byte 05 = color <PINK>
                                    byte 06 = color <RED>
                                    byte 07 = color <TURQUOISE>
                                    byte 08 = color <WHITE>
                                    byte 09 = color <YELLOW>
                                    byte 10 = attribute <BLINKING>
                                    byte 11 = attribute <COLUMN SEPARATOR>
                                    byte 12 = attribute <HIGH INTENSITY>
                                    byte 13 = attribute <NON DISPLAY>
                                    byte 14 = attribute <POSITION CURSOR>
                                    byte 15 = attribute <REVERSED IMAGE>
                                    byte 16 = attribute <UNDERLINE>
                                    byte 17 = attribute <PROTECT>
                                    byte 18 = <FIELD EXIT> required
                                    byte 19 = <EXIT RECORD> in effect
                                     byte 20 = <LOWER CASE> allowed
 040      AWHFLD    R               REFFLD(FIELD)
                                    blank when text
 050      AWHLIN    R               REFFLD(LINE)
                                    Row where to display
 053      AWHCOL    R               REFFLD(COLUMN)
                                    Column where to display
 056      AWHDLN    R               REFFLD(AWLEN)
```

```
                                    Edited length of field or text
059       AWHLEN    R               REFFLD(AWLEN)
                                    Unedited length of field or text
062       AWHDEC    R               REFFLD(NRDEC)
                                    Number of decimals
064       AWHVAR    R               REFFLD(AWHVAR)
065       AWHTX6    R               REFFLD(MESSAG)
```

```
       - text of 250 characters when text
*****************************************************************
DISPL   R AWHR07
 001      AWHT07    R               REFFLD(RECTYP)
                                    <07> = default value for previous field
 003      AWNX07    R               REFFLD(RECTYP)
                                    Record type of continuing record
                                    <00> = no continuing record
 005      AWHCO7    R               REFFLD(AWHCON)
                                    conditional display
                                    <0> = not, <1> = yes
 006      AWHIN7    R               REFFLD(INDUSE)
                                    byte 01      = more indicators active
                                    byte 02 - 10 = active indicators
 016      FILL7         49A         TEXT('FILLER')
 065      AWHTX7    R               REFFLD(MESSAG)
                  Text of default value
*****************************************************
DISPL   R AWHR08
 001      AWHT08    R               REFFLD(RECTYP)
                                    <08> = reserved for future use
 003      FILL81       250A         TEXT('FILLER')
 253      FILL82        62A         TEXT('FILLER')
*****************************************************************
DISPL   R AWHR09
 001      AWHT09    R               REFFLD(RECTYP)
                                    <09> = reserved for future use
 003      FILL91       250A         TEXT('FILLER')
 253      FILL92        62A         TEXT('FILLER')
```

# Appendix E : Collector Messages

The following are the Collector messages (error and diagnostic) reported in the `AWPRT` spool files.

## Collector Prepare Phase Messages

Messages from this phase are stored in the spool file AWPRT with user data AW310. The message meaning and action are described with each message.

| |
|---|
| **Object compiled using member in file1 in library1. Collector assumes library2**<br><br>**Action:** A compiled object was created based upon the source member member in file1 in library1. Either this source member does not exist in file file1 in library library1, or library library1 is not part of your library list. The Collector has assumed that the source member member found in file file1 in library library2 can be used as a suitable substitute.<br><br>**Meaning**: This can be a common and acceptable situation. In most cases the substituted source will be correct. You could perform one of the following actions : Add library library1 to your list of libraries if not already there. Re-run the Collector. Ignore the message and continue, assuming that the substituted library library2 contains the correct source member. Always verify this situation when in doubt. |
| **Object compiled using member in file1 in library1. Collector assumes in file2 in library1/2**<br><br>**Action:** A compiled object was created based upon the source member in file1 in library1. Either this source member does not exist in file1 in library1, or library1 is not part of your library list. The Collector has assumed that the source member member found in file2 in library library1/2 can be used as a suitable substitute.<br><br>**Meaning:** This can be a common and acceptable situation. In most cases the substituted source will be correct. You could perform one of the following actions : Add library library1 to your list of libraries if not already there. Re-run the Collector. Ignore the message and continue, assuming that the substituted file file2 in library library1/2 contains the correct source member. Always verify this situation when in doubt. |
| **Object compiled using member1 in file1 in library1. Collector assumes member2 in file1/2 in library1/2**<br><br>**Action:** A compiled DSPF object was created based upon the source member1 in file1 in library1. Either this source member does not exist in file1 in library1, or library1 is not part of your library list. The Collector has assumed that the source member2 found in file2 in library1 (or in library2) can be used as a suitable substitute. This error can only occur for DSPF source members generated by compiling a DSPF36 source.<br><br>**Meaning:** This can be a common and acceptable situation. In most cases the substituted source will be correct. You could perform one of the following actions : Add library library1 to your list of libraries if not already there. Re-run the Collector. Ignore the message and continue, assuming that the substituted file file2 in library library1 (or in library library2) contains the correct source member. Always verify this situation when in doubt. |

**Object object compiled using member member in file file in library library Member member does not exist in file file in library library. Check your list of libraries.**

**Action:** A compiled DSPF object was created based upon source member member in file file in library library. Either library library is not part of your library list or the library, file or member does not exist. The Collector could not find a suitable substitute in one of the libraries in the library list.

**Meaning:** This is an exceptional situation and should be resolved. The Collector requires the source member and object to extract texts. The collector can not extract screens without the corresponding source member. You should perform one of the following actions : Add library library to your list of libraries if not already there. Re-run the Collector. Compare the library list to the library list as used by the package to be processed and make sure that all libraries are included. Correct any inconsistencies and re-run the Collector. If no further action is taken then the source member will not be available for processing.

**Object object compiled using member member in file file in library library Member member does not exist in file file in library library Re-compile this member.**

**Action:** A compiled DSPF object was created based upon source member member in file file in library library. Either library library is not part of your library list or the library, file or member does not exist. The Collector found more than one source member in the library list which could be used as a substitute for the missing source member.

**Meaning:** This can be a common and acceptable situation. The Collector cannot be certain which of the source members will be used so requests that you recompile the source member. This will update the source member reference in the compiled object. You could perform one of the following actions : Recompile the source member and re-run the Collector. If no further action is taken then the source member will not be available for processing.

**No text or field found in the libraries you have entered**

**Action:** Either not a single DSPF was found in your list of libraries, or DSPF source members found were not treated due to one of the reasons mentioned in the log report.

**Meaning:** Check your library list or your log report.

**Object in Use.**

**Action:** The DSPF object (in object collect mode) was found to be in use. The object is skipped.

**Meaning:** It is likely the DSPF object is locked in use (by you or another user). Release the DSPF object and re-collect.

## Collector Extract Phase Messages

Messages from this phase are stored in the spool file `AWPRT` with user data AW800. Each message is preceded by the library, file and name of the member involved, and where applicable by the statement number containing the problem.

**Member member1 was encountered before in file file1 in library library1. Texts for this member are not treated or Object object1 was encountered before in library library1.**

**Action:** A source member or object occurs more than once in your library list.

**Meaning:** This can be a common and acceptable situation. The Collector assumes the first occurrence is the most recent one.

**Member has statements with duplicate or non-ascending sequence numbers.**

**Action:** This is only a diagnostic message, indicating the source member has its statement numbers in an unusual order.

**Meaning:** You may decide to edit the source member in order to re-sequence the statements on saving. This action will have no influence on the Collector.

**Member has deleted records. Texts for this member are not treated.**

**Action:** The Collector has encountered a source member containing one or more deleted records.

**Meaning:** This is a situation caused by handling the member in another way than editing (possibly a file utility like DFU). The Collector is not able to treat such a source member. You could perform one of the following actions: Edit the source member in order to re-sequence the statements on saving, thus eliminating the deleted record(s). Re-run the Collector. Ignore the message and continue, no pictures will be generated for this source member.

**Member is of type <DSPF36>. Texts for this member are not treated.**

**Action:** The Collector cannot handle source members having <DSPF36> syntax.

**Meaning:** You could perform one of the following actions : Re-compile the source member, thus creating its "native" DDS source member as well. Re-run the Collector. Ignore the message and continue, no pictures will be generated for this source member.

**No associated <*MENU> object found in your library list. Texts for this member are not treated.**

**Action:** The Collector has encountered a source member of type (MNUDDS) and can not find its object in your library list.

**Meaning:** The Collector is not able to treat such a source member as it can not distinguish its attributes like the presence of a key guidance and the form of the command line. You could perform one of the following actions : Re-compile the source member and re-run the Collector. Ignore the message and continue, no pictures will be generated for this source member.

**No associated <*FILE> object found in your library list. Texts for this member are not treated.**

**Action:** The Collector has encountered a source member of type <DSPF> and cannot find its object in your library list.

**Meaning:** The Collector cannot handle such a source member as it cannot distinguish attributes such as (referenced) field lengths. You could perform one of the following actions: Re-compile the source member and re-run the Collector. Ignore the message and continue, no pictures will be generated for this source member.

**Text found is too long, text is not extracted.**

**Action:** The Collector encountered a text of more than 999 characters.

**Meaning:** The Collector cannot handle texts of more than 999 characters. You could perform one of the following actions : Split up the text in smaller parts. Ignore the message and continue, an incomplete picture will be generated which should be processed manually.

**Field length greater than 999, field is not extracted.**

**Action:** The Collector encountered a field more than 999 characters.

**Meaning:** Collector cannot handle fields of more than 999 characters. You can only : Ignore the message and continue, an incomplete picture will be generated which should be processed manually.

**Variable found on (MSGID)-keyword, outputfield assumed.**

**Action:** The Collector encountered a (MSGID)-keyword and can not distinguish what message to use.

**Meaning:** This can be a common and acceptable situation. This message is only meant as information, there is no need to take any action.

**Unknown keyword found, starting with ..**

**Action:** The Collector encountered a syntax error in the source member.

**Meaning:** Check the source member.

**Unable to find valid syntax on ..**

**Action:** The Collector encountered a syntax error in the source member.

**Meaning:** Check the source member.

**Unable to find referenced <WINDOW>.**

**Action:** The Collector encountered a <WINDOW> keyword referring to another window, which on its turn cannot be found.

**Meaning:** Check the source member.

**Special edit code 5 to 9 used, edited length found in file <USREDT>.**

**Action:** The Collector encountered the use of the keyword EDTCDE(5), EDTCDE(6), EDTCDE(7), EDTCDE(8) or EDTCDE(9). The Collector is able to determine the edited length of the field.

**Meaning:** None.

**Special edit code 5/6/7/8/9 used, edited length not found in file <USREDT>**

**Action:** The Collector encountered the use of the keyword EDTCDE(5), EDTCDE(6), EDTCDE(7), EDTCDE(8) or EDTCDE(9) but cannot determine the edited length of the field.

**Meaning:** You could perform one of the following actions: Add a record to the file <USREDT> and re-run the Collector. Ignore the message and continue, an incomplete picture will be generated which should be processed manually.

**Length of 1 assumed for REFFLD(.....**

**Action:** The Collector encountered the use of the keyword <REFFLD>.The Collector cannot determine the length of the field, as it is not in the I/O-buffer for the display file and cannot be found in the referenced file defined by the <REFFLD> keyword.

**Meaning:** Normally this message will only apply to the use of the keyword <MSGID > together with the keyword <REFFLD>. MSGID's are not part of the I/O-buffer. You could perform one of the following actions: Inspect the <REFFLD> keyword (and possibly the <REF> keyword).

If the referenced file is not qualified (i.e. no fixed library name is defined), then add the library where the referenced file resides to your library list. Re-run the Collector. Set the length of the field in the source member (although a solution, this is not preferred). Ignore the message and continue, an incomplete picture will be generated which should be processed manually.

**Unable to find message XXXXXXX in message file YYYYYYYYY**

**Action:** The Collector encountered the use of a message, but did not find the messageID XXXXXXX in the message file YYYYYYYYY

**Meaning:** You could perform one of the following actions:

- Inspect your source member (a wrong ID could be coded)
- Inspect your message file
- Ignore the message and continue, an incomplete picture will be generated which should be processed manually.

**Object in Use**

**Action:** The DSPF object (in object collect mode) was found to be in use. The object is skipped.

**Meaning:** It is likely the DSPF object is locked in use (by you or another user). Release the DSPF object and re-collect.

**Related topics**

Collector Errors and Diagnostic Messages