

SuperReport Pro User Manual

Version 2.9

e-Node
30 rue de la République
33150 Cenon
France

<http://www.e-node.net>

Copyright and Trademarks

All trade names referenced in this document are the trademark or registered trademark of their respective holder.

SuperReport Pro is copyright Automated Solutions Group.

4th Dimension, 4D Compiler, 4D, 4D Server, 4D Client, and 4D Insider are trademarks of 4D SAS..

Windows is a trademark of Microsoft Corporation.

Macintosh, MacOS and MacOS X are trademarks of Apple, Inc.

Table of Contents

	Copyright and Trademarks.....	2
	Table of Contents	3
	Preface	8
	About this Manual	8
	Acknowledgments	8
Chapter 1:	Introduction	
	About SuperReport Pro	9
	SuperReport Pro Features	9
	What's New in SuperReport Pro 2.9.....	10
	What's New in SuperReport Pro 2.8.....	11
	System Requirements	12
	4th Dimension	12
	System Software	12
	Technical Support.....	12
	Registration	13
	License types.....	13
Chapter 2:	Installation	
	Plug-In Objects — Overview.....	14
	SuperReport Pro Plug-In	14
	Installation: Plug-In bundle (MacOS & Windows).....	14

Chapter 3: **Configuring SuperReport Pro**

- Declaring SuperReport Pro Variables 15
- SuperReport Pro Variables 15

Chapter 4: **SuperReport Pro Fundamentals**

- Creating SuperReport Pro Areas 17
 - Using as External Window 17
 - Using on 4th Dimension Form 18
- Customizing the SuperReport Pro Area 19
 - Customizing File Structure 19
 - Customizing User Access Variables 22
 - Customizing Standard SuperReport Pro Variables 23
 - Customizing SuperReport Pro Menu System 23
 - Customizing Report Editor 24
- Extending SuperReport Pro with Object Scripts 25
 - SuperReport Pro Script Limitations 26
- Understanding the SuperReport Pro Event Cycle 27
- Dealing with multi-platform Issues 28
 - Dealing with platform pathnames 28
 - Dealing with Print Drivers 29
- HTML Support 30
 - Using Custom Tag Variables 31
 - Understanding the SuperReport Pro Execution Cycle 31
 - Printing HTML Reports 32

Chapter 5: **Tutorial**

- Basic Integration 34
 - Using External Window 34
 - Using Plug-In Area 35
- SuperReport Pro Editor Introduction 38
 - Toolbar Overview 38
 - Section Marker Overview 40
 - Menu Overview 41
 - Creating a basic listing report 46
 - Creating a basic form report 48
 - Using Repeating Objects 49
 - Creating a break-processing report 50
 - Creating an HTML Report 59
 - Printing Arrays 67
 - Procedurally Creating Reports 70

Routine Overview	74
SR Pro – Access Routines	75
%SuperReport	76
SR Get Area.....	78
SR Set Area	79
SR Doc 2 Report.....	80
SR Load Report	81
SR Save Report.....	82
SR New Report	83
SR New Offscreen Area	84
SR DELETE OFFSCREEN AREA.....	85
SuperReport Pro — Customizing	86
SR Get Area Format	87
SR Set Area Format.....	88
SR Options	89
SR Get Options.....	90
SR Set Options.....	91
SR Std Vars.....	93
SR Structure	94
SR Variables.....	96
SR Commands	98
SuperReport Pro — Menus	101
SR Do Command.....	102
SR Menu Info	103
SR Menu Item	105
SuperReport Pro — Miscellaneous	107
SR ABOUT	108
SR Register	109
SR Document	111
SR File Types.....	112
SR Get Format Number.....	113
SR Get Table List.....	114
SR Get Field List	115
SR Get Commands As List BLOB	117
SR Get Tables As List BLOB	119
SR Get Fields As List BLOB.....	120
SR Get Structure As List BLOB.....	122
SR Get Variables As List BLOB	124
SR Get Variable Type	125
SR Get Virtual Variable	126
SR Get Indexed String	127
SR Is Modified	128
SR Main Table.....	129
SR Main Table2.....	131
SR On Event.....	132
SR Power Menu.....	136
SR RELATIONS	139
SR Set Modify.....	140

SR Get Tips Enabled	141
SR SET TIPS ENABLED.....	142
SR SWAP HANDLES (deprecated).....	143
SR Report To BLOB	144
SuperReport Pro — Printing.....	145
SR Get Number Of Pages.....	146
SR Page Setup	148
SR Preview	149
SR Print Disk.....	151
SR Print PICT	154
SR Print Report.....	156
SR Print HTML	158
SR Print To BLOB.....	160
SR Validate	162
SuperReport Pro — Developer Interface (API)	163
SR Set Editor Callback.....	164
SR Set Script Callback.....	167
SR Count Objects	170
SR Get Object IDs.....	171
SR Change Object Order.....	172
SR Reorder Objects	173
SR Create Object	174
SR Delete Object.....	178
SR Get Object Scripts.....	179
SR Set Object Scripts	180
SR Get Object Properties.....	181
SR Set Object Properties	184
SR Get Object Format.....	188
SR Set Object Format	190
SR Get Object Data	194
SR Set Object Data	195
SR Get Guides.....	196
SR Set Guides	197
SR Get Scripts	198
SR Set Scripts	199
SR Get Sections.....	200
SR Set Sections.....	202
SR Get Section Ids	203
SR Get Section Properties.....	204
SR Set Section Properties	206
SR Get Section Scripts.....	208
SR Set Section Scripts	209
SR Get Tool	210
SR Set Tool	211

Chapter 7: **SuperReport Pro Codes**

File Manager Error Codes	215
Section Codes.....	216
Error Codes	217
Event Codes	218

Menu Numbers.....	219
Editor Codes — Action Types	222
Editor Codes — Object Types	222
SuperReport Pro Constants.....	223

Chapter 8:

Hints & Tips

Troubleshooting.....	227
SuperReport Pro — Wrapper Routines.....	212

Preface

SuperReport Pro has been developed to provide an enhanced reporting creation and printing tool for developers and end-users. Using the SuperReport Pro Editor and plug-in routines, developers can create sophisticated reports that support a variety of data formats, including 4th Dimension fields, variables, and arrays.

In addition, SuperReport Pro can be integrated into any 4th Dimension application's user interface, providing your users with the ability to create reports using a number of advanced features not available using the standard reporting tool built into 4th Dimension.

About this Manual

Throughout this manual, you will see various formatting options to better distinguish the use of plugin calls, 4th Dimension commands, and 4th Dimension methods/functions. Listed below is a sample of each reference used.

All examples in this documentation are based on 4th Dimension v6. If you are using 4th Dimension v3, the examples in this documentation will vary.

4D Commands and Functions	ALERT
4D Methods	<i>SuperReport</i>
4D Plugin calls	SR Print Report
4D Variable	SRDate
Code Examples	<i>SuperReport(->[Customers])</i>

Acknowledgments

Original Design and Programming by **John Sidney-Woollett** and **Michael S. Erickson**

Contributions by Stewart Buskirk, Miloslav Bystricky, and Aparajita Fishman

Documentation by Michael S. Erickson & Philippe Ganter

Published and distributed by e-Node

SuperReport Pro is a trademark of Automated Solutions Group

1 — Introduction

SuperReport Pro™ consists of a plug-in area, which is placed on your input form or dialog and a group of supporting plug-in routines, which can be used to procedurally interact with report templates. You can implement SuperReport Pro by simply placing the report plug-in area on your form, and provided additional customization using the plug-in routines.

About SuperReport Pro

SuperReport Pro's greatest benefit is the fact that it provides the ability for users to create and modify report designs long after a 4D database application has been completed without requiring any code changes to the database. This benefit is further enhanced in the case of compiled databases - where SuperReport Pro allows reports to be changed without requiring either database changes or recompilation.

SuperReport Pro reports can be held as documents on disk, or be contained within the data file belonging to the database. Thus vertical market applications sold to many customers can be supplied with standard reports, which can later be customized by the user. Since these reports are held in the data file, program updates installed at a later date will have no impact on the different users' systems - the same program update can be sent to all users and all the user's customized reports are preserved!

SuperReport Pro's power does not come at the expense of simplicity. Most users will be able to grasp the concepts involved in designing reports since the editor is so similar to a drawing program. Report operations like break processing which can be cumbersome and complex in 4D are also made simpler, with the user able to specify when breaks occur, and which objects are to be totaled without requiring any code.

SuperReport Pro Features

SuperReport Pro includes a wide variety of features for creating fully customizable reports. Using the full-featured plug-in area, you can quickly and easily enhance your applications reporting capabilities.

If the standard functionality is not enough, you can take advantage of the complete developer API to further extend the power of SuperReport Pro!

Included in SuperReport Pro™ are the following features:

- Compatibility with 4D 2004 and above, including v11 SQL and v12
- Include a full featured reporting tool in any application, quickly and easily
- Enhance reporting capabilities using the SuperReport Pro developer API
- Complete support for creating HTML formatted reports
- HTML Report Generation Support
- Repeating objects for print related many records
- Supports printing 4th Dimension arrays, including direct access to specific array elements
- Support for executing callback routines and object scripts when printing reports
- Support for execute custom script execution systems for enhanced script management
- Support for custom structure views – great for isolating what tables and fields user can access
- Developer API for customizing end-user functionality
- Enhanced Editor Customization

What's New in SuperReport Pro 2.9

- **The storage of SuperReport Pro reports changes from picture to blob**

In 4D v11 and v12, it is no longer possible to use picture fields to store SuperReport Pro reports. All SuperReport Pro calls accepting picture parameters are changed to blob parameters. However, it is not possible to just change the field type in 4D from picture to blob, report stored in picture fields must be converted to blob fields with the new SuperReport Pro method **SR Report To BLOB**.

This command accepts a SuperReport Pro report in picture parameter and returns a blob:

SR Report To BLOB (picture) -> blob

The older call **SR SWAP HANDLES** is deprecated and should not be used in 4D v11 and v12.

It is recommended to use **SR Report To BLOB** in 4D 2004 (before data conversion) to convert SuperReport Pro reports from picture field to blob field. However, if the data file is already converted, it is still possible to use the call to convert the reports and their scripts.

The recommended sequence is as follows:

1. if there are SuperReport Pro reports stored in picture fields in 4D 2004 database, create a blob field for each picture field
2. use call SR Report To BLOB to move reports from picture fields to blob fields in 4D 2004
3. convert the data to 4D v11/v12
4. in v11/v12, use only blob fields

It is possible to export all reports from picture fields in 4D 2004 and reimport them to blob fields in 4D v11/v12 as well.

Note that as parameter declaration of SuperReport Pro calls is changed from Picture to Blob, compiler in 4D will show all places where picture variable or field are passed to SuperReport Pro as compiler errors.

- **SuperReport Pro is now fully compatible with 4D v11/v12**
- **Fixed Print Preview on MacOS X**
- **Fixed script editing on Windows (CR/LF handling)**
- **Removed Cancel button from SuperReport Pro Preview button**

In 4D v11/v12 a plugin cannot close its window in which it displays a plugin area.

- **Fixed a problem where SuperReport Pro tried to display a dialog when the Preview window was displayed**

This resulted in a freeze of 4D

- **e-Node license system**

Only one registration key is required. The key is either linked to the 4D or 4D Server serial number, or to the name of the company / developer. Please refer to the Registration section on page 13.

What's New in SuperReport Pro 2.8

- **Enhanced User Interface**

SuperReport Pro has been enhanced, providing a revamped user editor interface. All interface elements are now consistent with the internal 4th Dimension interface and utilizes modern interface elements.

- **Developer Interface**

SuperReport Pro now includes a full-featured developer interface which provides 4th Dimension application developers will a suite of routines for procedurally creating and managing reports, as well as routines for customizing the entire Editor interface.

- **Enhanced HTML Interface**

SuperReport Pro includes new interfaces for creating and working with reports from within web applications. Developers can now create reports using SuperReport, which may be invoked and displayed within web applications.

- **New and Enhanced Commands**

SuperReport Pro includes over 50 new and enhanced commands, providing more power and flexibility for report creation and management requirements.

- **Report Manager**

Using the SuperReport Pro Report Manager, users will be able to manage all the various reports for your application. Since the Report Manager is based on standard 4th Dimension code, you can modify the Report Manager system to suit your needs, or use it as is.

- **Procedural PDF Output (MacOSX Only)**

SuperReport Pro 2.8 includes new routines for procedurally creating PDF reports when running on MacOSX. These routines use the OSX native PDF libraries.

System Requirements

4th Dimension

SuperReport Pro is compatible with 4th Dimension v2004 and higher, or 4D Server v2004 or higher, including 4D v11 SQL and v12.

System Software

SuperReport Pro is compatible with Windows XP or greater and Mac OS 10.4 or greater.

Technical Support

Technical support for SuperReport Pro will be provided electronically via e-mail or our online support reporting system. You are encouraged to use the online web reporting form as it will be correctly routed to the appropriate support personnel.

<http://www.e-node.net>

Registration

SuperReport Pro requires a registration key to “unlock” the product making it a full working version. Call the **SR Register** command in the On Startup method.

Without the registration key, SuperReport Pro will operate in demonstration mode during 20 minutes.

Version 2.9 introduced a new license design. Previous licenses will not work with this release.

In order to activate SuperReport Pro 2.9 and above, you need to require a new license key from e-Node.

License types

Like all e-Node plug-ins, SuperReport Pro offers six different license types. There are no such things as MacOS vs Windows or Development vs Deployment:

Single user license. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on 4D Standalone or Runtime. Since the registration key is linked to a specific 4D license, you need to provide the number returned by the 4D command **GET SERIAL INFORMATION** (first parameter). A new license will be supplied for free at any time if you change your 4D version and / or get a new 4D registration key, provided that your previous licenses match the current public version at the exchange time.

Small server. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on 4D Server up to 10 users. The registration key is linked to your 4D Server license just as above.

Medium server. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on 4D Server up with 11 to 20 users. The registration key is linked to your 4D Server license just as above.

Large server. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on 4D Server over 20 users. The registration key is linked to your 4D Server license just as above.

Unlimited Single User. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on as many 4D Standalone, Runtime or Engine copies that run your 4D application(s). This is a yearly license, which expires after the date when it is to be renewed. The expiration only affects interpreted mode. **Compiled applications using an obsolete license will never expire.**

Unlimited OEM. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on as many 4D Server (of any number of users), 4D Standalone, Runtime or Engine copies that run your 4D application(s). This is a yearly license, which expires after the date when it is to be renewed. The expiration only affects interpreted mode. **Compiled applications using an obsolete license will never expire.**

A 4D database used to retrieve your 4D serial information is available from the following link:

<http://www.e-node.net/ftp/GetSerialInfo>

2 — Installation

This chapter outlines the steps necessary for installing SuperReport Pro into your existing applications. When installing SuperReport Pro, you must have access to the source code version (4th Dimension plugins cannot be installed into compiled applications).

Plug-In Objects — Overview

SuperReport Pro Plug-In

The SuperReport Pro plug-in contains the plug-in area which is placed on the desired 4th Dimension form in which you wish to provide user access, as well as all the supporting routines which interact with the SuperReport Pro plug-in area.

Installation: Plug-In bundle (MacOS & Windows)

SuperReport Pro is provided as a plug-in bundle for 4D 2004, 4D v11 SQL, 4D v12 or higher.

This single version will work with MacOS and Windows deployments (you don't need separate MacOS and Windows versions).

- 1 — Locate the folder where SuperReport Pro has been installed on your computer.
- 2 — Locate the 4th Dimension structure where you wish to install the SuperReport Pro plug-in.
- 3 — If you don't already have a directory labeled "Plugins", create one now.
- 4 — Copy the following plug-in to your applications Plugins folder: SuperReport Pro.bundle.

3 — Configuring SuperReport Pro

This chapter outlines the steps necessary for configuring SuperReport Pro in your custom applications. Since SuperReport Pro is a combination of 4th Dimension procedures, layouts and external routines, it is important that these steps be completed before using SuperReport Pro.

For additional information on using SuperReport Pro, please refer to the **Chapter 5 - Tutorial** as it provides a variety of examples on using SuperReport Pro.

Declaring SuperReport Pro Variables

In addition to the various plug-in routines, SuperReport Pro contains a set of variables, which are provided as a method for communicating with your code. Each of these variables must be properly declared and initialized in process in which you are using SuperReport Pro.

Step 1: Add the following code segment to each routine that is started via the **New Process** command in which you wish to use SuperReport Pro.

If you are not using the **New Process** command to display the forms, which contain the SuperReport Pro plug-in area, you should declare these variables in your applications *On Startup* method.

C_DATE(SRDate) `contains the current date
C_TIME(SRTime) `contains the current time
C_LONGINT(SRPage) `contains the current page number
C_LONGINT(SRRecord) `contains the current record number or iteration

SuperReport Pro Variables

There are a number of internal variables used by SuperReport Pro. The following is a description of each variable and their usage during reporting operations.

If you wish to replace the default SuperReport Pro variables, providing a more descriptive representation for users as it relates to your application, you can use the **SR Std Vars** routine.

In addition, each of these variables can be used in any callback procedure, which is executed by SuperReport Pro during a printing operation.

<u>Variable Name</u>	<u>Data Type</u>	<u>Description</u>
SRArea	C_LONGINT	Current report area reference (can be used in callback scripts)
SRDate	C_DATE	Contains the current date
SRTime	C_LONGINT	Contains the current time.
SRObjID	C_LONGINT	Contains the current object for which you have a script attached
SRPage	C_LONGINT	Contains the current page
SRRecord	C_LONGINT	Contains the current record number or iteration value

4 — SuperReport Pro Fundamentals

This chapter outlines the fundamentals of using SuperReport Pro, including:

- An overview of creating SuperReport Pro areas
- Customizing the SuperReport Pro area
- Extending SuperReport Pro with object scripts
- Understanding the SuperReport Pro Event Cycle
- Dealing with multi-platform issues.
- HTML Support

For detailed information on using the SuperReport Pro commands, please refer to **Chapter 6 - Command Reference**.

Creating SuperReport Pro Areas

There are two different ways in which a SuperReport Pro area can be presented to your users.

- Using the 4th Dimension Open external window command
- Displaying a 4th Dimension form (either via a dialog or input form)

Using as External Window

If you wish to use SuperReport Pro in an external window, you simply need to call the 4th Dimension Open external window routine.

C_LONGINT(iSR_Win)

iSR_Win:=**Open external window**(50;50;400;400;8;"SuperReport Pro";"%SuperReport")

Using on 4th Dimension Form

The most common way of using SuperReport Pro is to place a plug-in object on your input form or dialog. Simply create a plug-in object on your input form, select the %**SuperReport** item from the Routine popup menu and you are ready to start using SuperReport Pro.

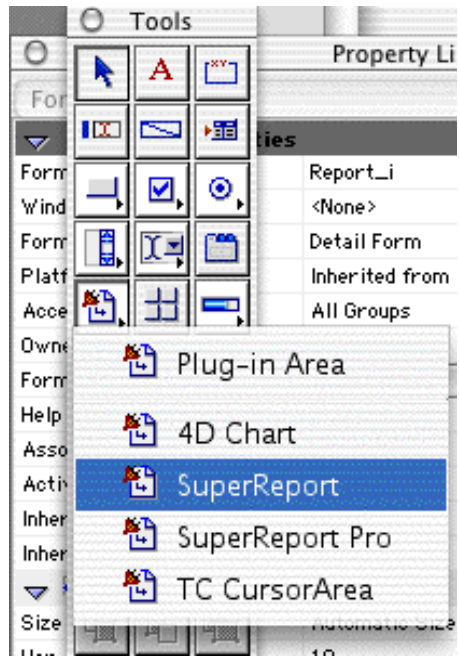


Figure 1a — SuperReport Pro plug-in area in Object Properties Palette

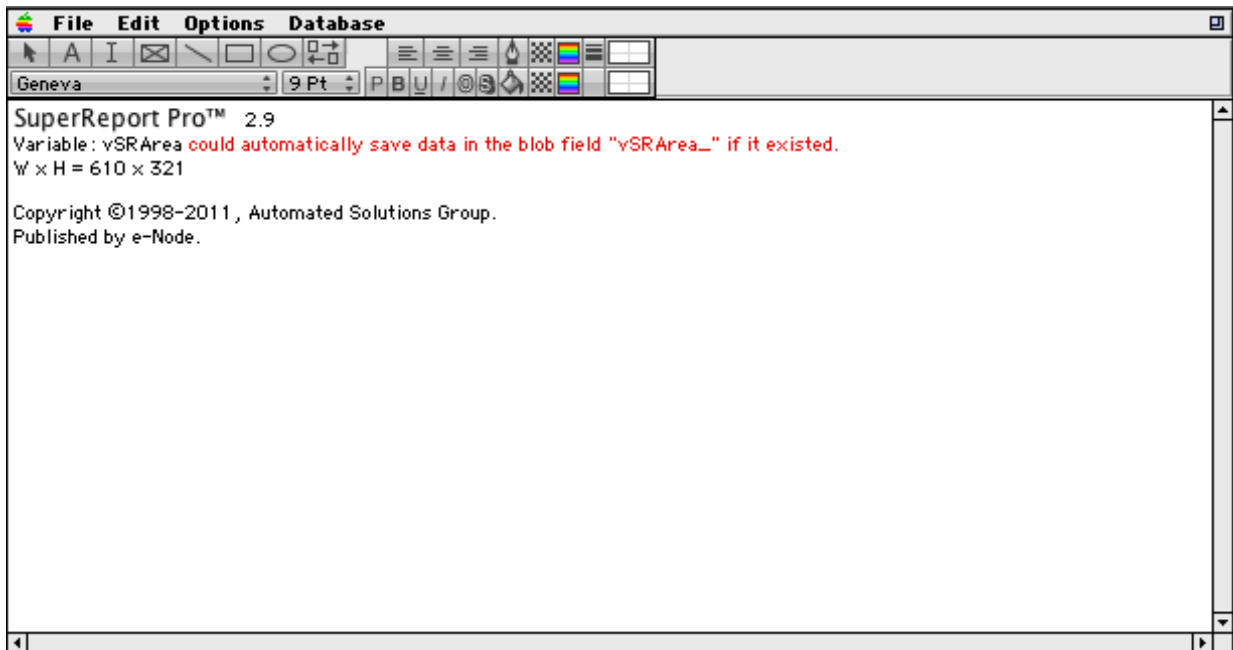


Figure 1b — SuperReport Pro as plug-in area on a 4D form

For more detailed information on using SuperReport Pro on an input form, please refer to the **Chapter 5 – Tutorial**.

Customizing the SuperReport Pro Area

There are a variety of ways you can customize the SuperReport Pro area to suit your applications needs.

- Customize the table and field access using **SR Structure**
- Customize the variable list using **SR Variables**
- Customize the command list using **SR Commands**
- Customize the SR Pro standard variable using **SR Std Vars**
- Customize the appearance of the Report Editor using **SR Options**
- Customize the menu system using **SR Menu Item**

Customizing File Structure

Perhaps one of the most commonly used routines in SuperReport Pro is the SR Structure command. Using this routine, you can customize the file structure so that only those tables and fields you wish to provide your users access will be display.

In addition, you can also provide a table or field alias to better signify what the field contains without having to modify the actual 4th Dimension file structure.

Using 4Ds Virtual Structure

SuperReport Pro supports customized structures, which were created using 4th Dimension routines **SET TABLE TITLES** and/or **SET FIELD TITLES**.

Unlike the SuperReport Pro routine SR Structure, which must be called each time the report form is opened, 4Ds Virtual Structure will remain active until another call to **SET TABLE TITLES** and/or **SET FIELD TITLES**.

The following example which customize the structure to only show the Customer, Invoice, and Item tables (see similar example for SR Structure interface).

```
ARRAY LONGINT($aiTableNo;3)  
ARRRAY STRING(32;$asTableName;3)
```

```
$aiTableNo{1}:=Table(->[Customer])  
$aiTableNo{2}:=Table(->[Invoice])  
$aiTableNo{3}:=Table(->[Item])
```

```
$asTableName{1}:=Table name(->[Customer])  
$asTableName{2}:=Table name(->[Invoice])  
$asTableName{3}:=Table name(->[Item])
```

```
SET TABLE TITLES($asTableName;$aiTableNo)
```

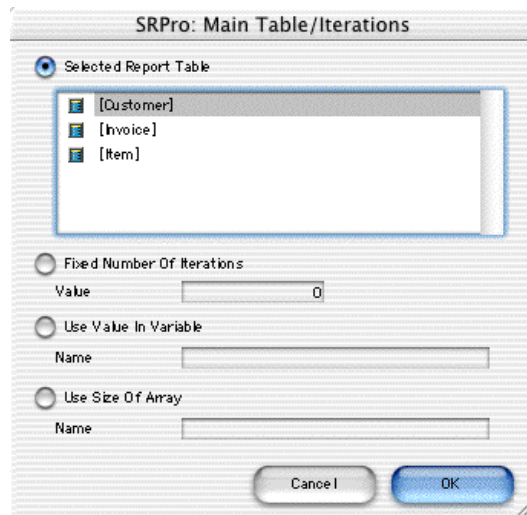


Figure 2 — SuperReport Pro Table Selection Dialog

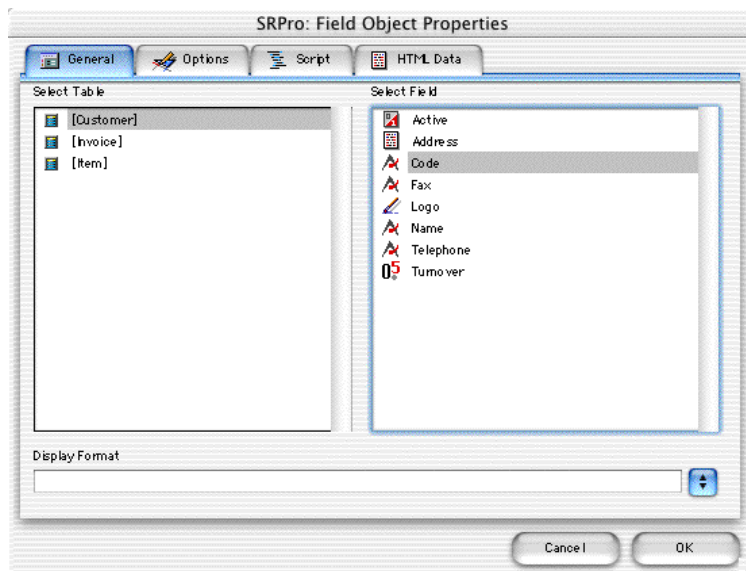


Figure 3 — SuperReport Pro Field Selection Dialog

Using SR Structure

Another option of display a customized structure is to use the SuperReport Pro command SR Structure. This routine has existed since the first release of SuperReport Pro and is left for backwards compatibility with applications, which already have SuperReport Pro, installed.

If you don't call **SR Structure** with the custom file structure, SuperReport Pro will display the default file selection (typically all visible tables and fields). For more details on using SR Structure, please refer to **Chapter 6 -Command Reference**.

In your report forms On Load event, add the following code. All structure customization must be called before calling **SR Set Area**, otherwise SuperReport Pro will not function correctly.

Enter the following code:

```
ARRAY STRING(63;aSR_Structure;6)
aSR_Structure {1}:="1;3;0;[Customers]"
aSR_Structure {2}:="1;3;0" `add all of [Customer] file
aSR_Structure {3}:="2;4;0;[Invoice]"
aSR_Structure {4}:="2;4;0" `add all of [Invoice] file
aSR_Structure {5}:="3;5;0;[Item]"
aSR_Structure {6}:="3;5;0" `add all of [Item] file

$result:=SR Structure (eReportArea;" aSR_Structure ")
```

Customizing User Access Variables

Like the SuperReport Pro structure system, you can also customize the SuperReport Pro variables by calling the SR Variables routine.

If you don't call SR Variables, no variables will be available and the variable popup item will be disabled. For more details on using SR Variables, please refer to Chapter 6 - Command Reference.

In your report forms On Load event, add the following code. All variable configuration must be called before calling **SR Set Area**, otherwise SuperReport Pro will not function correctly.

Enter the following code:

```
ARRAY STRING(63;aSR_Variables;5)
aSR_Variables {1}:="1;Report Variables"
aSR_Variables {2}:="1;Report Date;SRDate;1"
aSR_Variables {3}:="1;Report Time;SRTime;1"
aSR_Variables {4}:="1;Page Number;SRPage;1"
aSR_Variables {5}:="1;Record Number;SRRecord;1"
```

```
iSR_Err:=SR Variables (eReportArea;"aVars")
```

When the SuperReport Pro editor is displayed, all places where a variable selection popup menu is displayed, the following menu structure will be used:

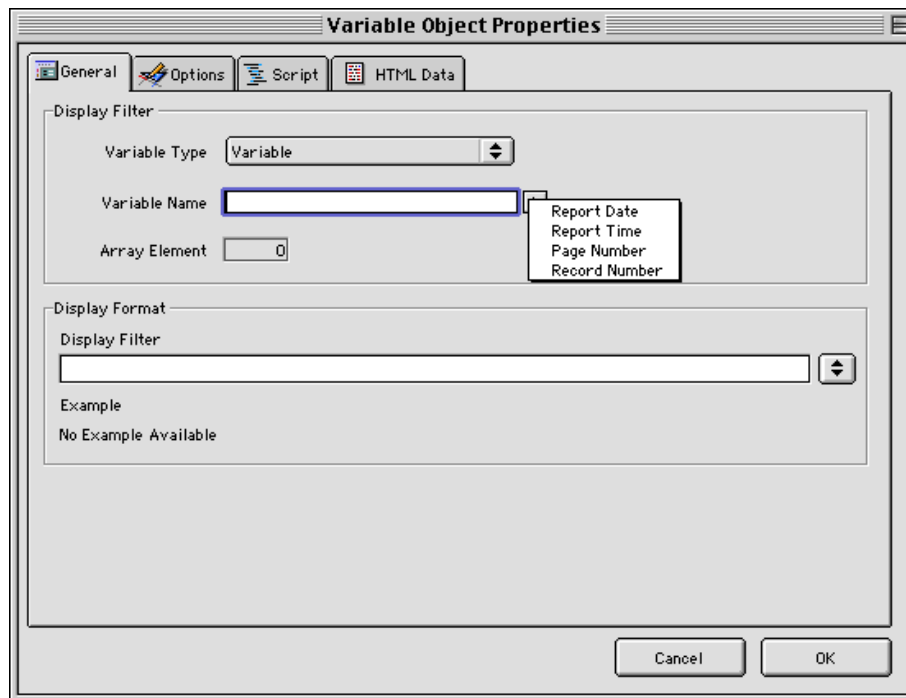


Figure 4 — SuperReport Pro Variable Selection Dialog

Customizing Standard SuperReport Pro Variables

The standard SuperReport Pro variables (SRDate, SRTime, SRRecord, and SRPage) can be overridden using your own variable names, providing further report editor customization. For more information on customizing the standard SuperReport Pro variables, please refer to **Chapter 6 - Command Reference**.

The following example changes the SRDate variable to use a custom date variable labeled “dMyDateVar”

C_DATE(dMyDateVar) `make sure you declare the variable to the correct data type

```
iSR_Err:=SR Std Vars(1;"dMyDateVar")
```

Customizing SuperReport Pro Menu System

The final portion of the SuperReport Pro Editor, which can be customized, is the standard menu system. You can perform a variety of menu item modifications, including:

- Set the enabled/disabled attribute for a given menu item
- Set the mark character for a given menu item
- Provide an override method for a given menu item

For example, if you want to present your own Query Editor (such as QueryPack™) when the Select Records... menu item is selected, you could supply a replacement 4th Dimension method that will be executed instead of the default action when the menu item is selected.

The following code will configure SuperReport Pro to display a custom 4th Dimension method “myQuery” when the Select Records... menu item is selected:

C_LONGINT(iSRProMenu)

```
iSRProMenu:=402 `Select Records menu number
```

```
iSR_Err:=SR Menu Item(eReportArea;1;iSRProMenu;"";0;0;"myQuery")
```

Customizing Report Editor

The last item of customization we'll cover is the SuperReport Pro Editor. Using the **SR Set Options** command, you can customize a variety of items, which affect the way the SuperReport Pro Editor is displayed.

In this example, we'll only cover a few of the customization options available using the SR Options command. For more details on using **SR Set Options**, please refer to **Chapter 6 - Command Reference**.

In your forms On Load event, call the **SR Set Options** routine, supplying the SuperReport Pro area reference and the desired customization options.

The following example will disable access to scripts.

```
C_BLOB(pReportData)
```

Case of

```
:(Form event=On_Load)  
  iSR_Err:=SR New Report(pReportData)  
  iSR_Err:=SR Set Area(eReportArea;pReportData)  
  iSR_Err:=SR Options(eReportArea;-1;1;-1)
```

End case

The following example will disable the zoom window, and disable access to HTML buttons.

```
C_BLOB(pReportData)
```

Case of

```
:(Form event=On_Load)  
  iSR_Err:=SR New Report(pReportData)  
  iSR_Err:=SR Set Area(eReportArea;pReportData)  
  iSR_Err:=SR Options(eReportArea;-1;-1;1+4) `1 - hide zoom; 4 - disable HTML
```

End case

Extending SuperReport Pro with Object Scripts

Like standard 4th Dimension reports created using the Form Editor, SuperReport Pro objects may contain scripts which may call any 4th Dimension command, method, or 3rd Party plug-in routine. Using SuperReport Pro scripts, you can further customize your reports to produce output, which cannot be created using standard SuperReport Pro objects.

For example, let's say you wanted to create a report which a listing of sales, broken down by customer, and provide an average sale for each custom as well as an overall invoice average. While SuperReport Pro can handle most of these tasks for you, utilizing the advanced break processing capabilities built-in to SuperReport Pro, the extended averaging requirements would not be possible without the support of object scripts.

Using SuperReport Pro's object scripts, you can place custom code on just about any SuperReport Pro object, including section lines, providing a very complete and flexible reporting system.

The following is the standard SuperReport Pro Script Editor. Using the editor, you can enter any valid 4th Dimension command, method, or plug-in routine.



Figure 5 — SuperReport Pro Script Editor

SuperReport Pro Script Limitations

While SuperReport Pro provides the ability to easily create custom scripts, there are a few limitations that you should be aware of when creating your scripts.

1. SuperReport Pro scripts cannot have standard 4th Dimension conditional commands such as For/End for, Case/End case, While/End while. If you need to perform conditional checks while printing a SuperReport Pro report, you must call a custom 4th Dimension method that in turn calls the conditional code.
2. SuperReport Pro scripts are not executed in compiled form, regardless of your applications compiled status. While this typically won't be a noticeable issue, you should be aware that scripts are slower than compiled code.
3. All scripts are executed in the same fashion as call the 4th Dimension **EXECUTE** command.
4. When selecting a table or field, which has been customized (using SuperReport Pro's structure customization feature), the script editor will display the actual table/field information as configured at the structure level.

Understanding the SuperReport Pro Event Cycle

Due the extensive reporting features offered by SuperReport Pro, there is certainly some level of potential confusion for even the seasoned 4th Dimension programmer. The following information outlines the standard SuperReport Pro Event Cycle.

SuperReport Pro generates reports in the following sequence (in pseudo-code):

Execute StartProcedure

Determine how many iterations are required

```
While (Current Iteration < Maximum Iterations) do
  Execute BodyProcedure
  If (Break Values have changed)
    Process appropriate SubHeader sections
    Process appropriate SubTotal sections
  End if
  Process Body section
End while
```

Process Total section

Execute EndProcedure

Section Processing

In processing any section, the following sequence of events takes place:

```
Execute section script
For (i = 1 to number of objects in this section) do
  Execute object script
  Get object (field/variable/array) value
End for
```

Dealing with multi-platform Issues

Fortunately, SuperReport Pro does a good job of conditionally handling most multi-platform issues. However, there are some cases when some additional developer control is necessary to support multi-platform issues.

- Dealing with platform pathnames
- Dealing with Print Drivers

Dealing with platform pathnames

All Macintosh files use a directory delimiter of “:” (ASCII 58), while Windows directories are delimited with a “\” character. One approach to dealing with these differences is to use a 4th Dimension variable, which contains the delimiter.

```
$delim:=Char(58) `default, Macintosh delimiter
If(IsWindows)
  $delim:=Char(92)
End if
```

Then you could use the 4th Dimension function Application file or Structure file to return the pathname to either the application or structure file as starting point to determine the pathname to a “Reports” directory.

For some useful routines for obtaining the parent pathname for an application or structure file, refer to the 4th Dimension Language Reference.

```
C_TEXT(tStructPath;tReportPath)
```

```
tStructPath:=TC_HFSParentName(Structure file) `TC_HFSParentName is from Toolchest from ASG
tReportPath:=tStructPath+”Reports”+$delim+”myReport.srp”
iSR_Err:=SR Save Report(pSR_Report;tReportPath)
```

Dealing with Print Drivers

Due the large number of print drivers available for Windows (unlike the standard LaserWriter driver available for Macintosh), configuring reports for most printers can be a very difficult task. The following is a list of some tips which can make designing your reports a manageable job.

1. Try to leave at least 1/4" margin on the edges of your reports to account for wide variety of print drivers
2. Stick with common fonts, most Windows applications have fonts such as Arial or Times Roman.
3. Be careful of using enhanced font formatting attributes such as Bold, Italic, Underline, etc. While most print drivers can handle multiple styles for a single object, some print drivers have problems dealing with multiple styles.
4. Test, test, test! Try to test your reports on as many different printers as possible... you would be surprised at the differences between drivers.

If you have a tip that would be useful to other SuperReport Pro users, please let us know so that we can post this information in future documentation and release notes, as well as our web site.

Send tips to: srp@e-node.net

HTML Support

SuperReport Pro provides the ability to create standard HTML reports.

Each SuperReport Pro object may contain custom HTML tags, which are inserted before and after the corresponding data object is sent to the HTML file when printing the report in HTML format.

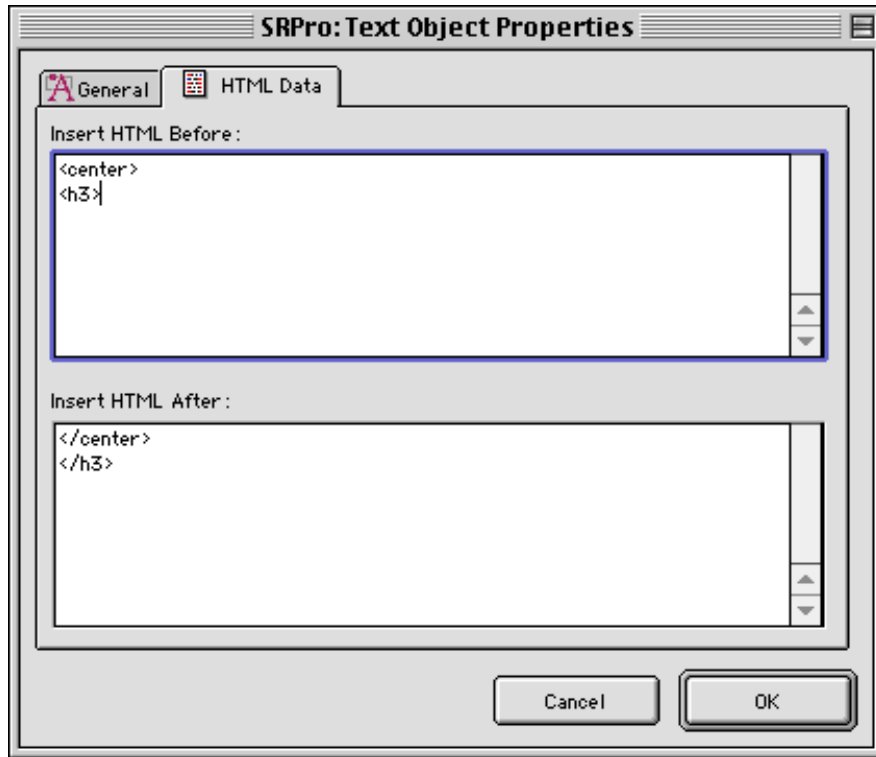


Figure 6 — SuperReport Pro HTML Dialog

Using the HTML Tag Dialog, you can supply the following items:

- **Begin Tag**
The begin tag information is sent to the HTML file before the corresponding data is sent to the file.
- **End Tag**
The end tag information is sent to the HTML file after the corresponding data is sent to the file.

Using Custom Tag Variables

In addition to entering static HTML tags in the HTML Dialog, you can also imbed custom HTML tag variables, which are actually 4th Dimension variables which may contain any valid HTML tag (or group of tags).

For example, if you have a set of HTML tags which you insert at the beginning of all your HTML files, you could place all the HTML data into a 4th Dimension variable, then reference the variable from with SuperReport Pro by using the custom variable tags.

To use a 4th Dimension variable within your HTML tags, enclose the variable with “<%” and “%>” markers. When the report is printed, SuperReport Pro will take the data between these markers, retrieve the data from the actual variable and insert it into the HTML stream as the file is created.

The following is an example 4th Dimension variable that is inserted using the variable tags.

```
tBegTagHTML:="<html><body bgcolor='#ffffff'>" `sample start file tag  
tEndTagHTML:="</body></html>" `sample end file tag
```

Understanding the SuperReport Pro Execution Cycle

When creating HTML reports, it is important that you understand the 4th Dimension execution cycle as this is how you will determine when specific tags should be include so that you can correctly create your HTML file.

For more information on the SuperReport Pro Execution Cycle, please refer to the **Understanding the SuperReport Pro Event Cycle** earlier in this chapter.

Printing HTML Reports

There are three different ways in which you can print your reports in HTML format. The first method uses the SuperReport Pro Editor.

Step 1: Select **Print To Disk...** for the **File** menu.

Step 2: Select **Output As HTML** for the **Output Options** section.



Figure 7 — SuperReport Pro Print To Disk Dialog

The second method is to use the SR Print HTML command. For more information on the options available when call SR Print HTML, please refer to **Chapter 6 - Command Reference**.

\$err:=SR Print HTML(pSR_Report;"report.html";0; SR All Sections)

The third method is to use the SR Print BLOB command. For more information on the options available when call SR Print BLOB, please refer to **Chapter 6 - Command Reference**.

C_BLOB(oSR_Report)

SET BLOB SIZE(oSR_Report;0)

\$err:=SR Print To BLOB(pSR_Report;oSR_Report; SR PrintToBLOB Output HTML; SR All Sections;9;13)

5 — Tutorial

Well, now that you have completed the installation and configuration process, it is time to begin using SuperReport Pro. This section provides various examples of how SuperReport Pro can be used in your applications.

SuperReport Pro provides an extensive feature set for providing reporting interfaces for your custom applications and your end users. Whether you purchased SuperReport Pro to create custom reports, or you wish to provide the ability for your users to create and maintain their own reports, SuperReport Pro can handle the job.

This chapter outlines a variety of ways in which you can use SuperReport Pro to provide advanced reporting capabilities within your application.

Basic Integration

SuperReport Pro can be integrated into your applications in two different methods. The first method uses the 4th Dimension Open external window, the second uses a 4th Dimension form that can be displayed in a dialog or input form.

All examples discussed in this section assume you have a custom table for storing the various reports you create. In our example, we'll have the following table structure:



SR_ReportManager	
SR_ReportName	
SR_Description	
SR_ReportData	
SR_TableNo	

Figure 8 — Report Manager Structure

Using External Window

Using the 4th Dimension Open external window command, you can display the SuperReport Pro editor in a fashion similar to the **New process** command.

Step 1: Open the SuperReport Pro editor in an external window using the following code.

```
C_LONGINT(iSR_Win)
```

```
iSR_Win:=Open external window(50;50;Screen width-50;Screen height-50;8;"New Report";"%SuperReport")
```

When you are display SuperReport Pro in an external window, you should use a Window type of 8 so that the window can be resized by the user and the close box is available. If you use a modal window (type 1 or type 5), the user will not be able to close the window.

Step 2: If you wish to communicate with the SuperReport Pro editor in an external window, you can use the window reference (iSR_Win) as the area reference parameter for those routines which interact with the plug-in area.

```
C_TEXT(tReportDoc)
```

```
tReportDoc:="Hard Disk:myReport.srp"
```

```
iSR_Err:=SR Load Report (iSR_Win;tReportDoc)
```

Using Plug-in Area

The second (and most common) way to use SuperReport Pro is to place the plug-in area on an input form, then provide access to the input form, either via the DIALOG command or through the standard MODIFY SELECTION command (or User Mode access).

- Step 1:** Create a new input form or modify an existing input form, which you wish to display the SuperReport Pro plug-in area.
- Step 2:** Create a new plug-in object using the 4th Dimension **Property List** (Figure 9).
- Step 3:** Select the **SuperReport** object from the Plug-in Area popup menu to create a SuperReport Pro area.

Note: The SuperReport Pro object is displayed as it is used when using SuperReport Pro in 4D's user environment

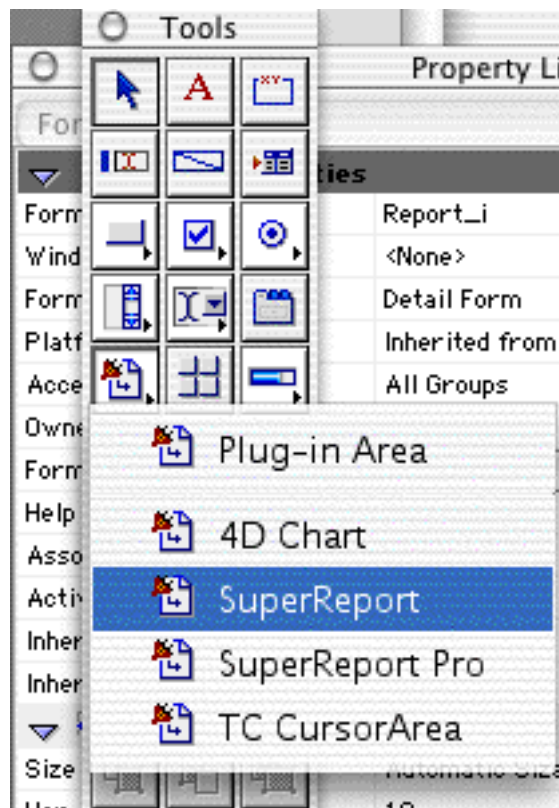


Figure 9 — Object Properties Palette using SuperReport area

Step 4: Assign a name to your variable, which will be used by the related SuperReport Pro command to reference your report area.

If you define the name of the report area which is the same name followed by an underscore of a field which is in the table which contains the form you have created, SuperReport Pro will automatically load and save the report when the record is loaded.

For example, if you have a field entitled [Reports]ReportData_ when creating the SuperReport Pro area, using a name of “ReportData” will instruct SuperReport Pro to automatically load/save the report when the record is loaded and the form is displayed.

If you are using the auto save feature of SuperReport Pro, you may proceed to Step 7.

If you are not using a “report” table, you will need to procedurally load/save the reports using the appropriate SuperReport Pro routines (Step 5 and Step 6).

Step 5: The next step is to create the report data so that your users can access it. The first If test determines if we are access an existing record. If so, we load the data from the saved report record (we’ll cover how to save report data in a bit); otherwise we create a new report using the **SR New Report** command.

C_BLOB(pSR_Report) `used by SR Get Area and SR New Report
C_LONGINT(iSR_Err)

Case of

:Form event=(On Load)

If (Record number([Report_Mgr]) # Is new record) `if this is not a new record

\$result:=SR Set Area (eReportArea;[Report_Mgr]ReportData)

Else

\$result:=SR New Report(pSR_Report) `create a new report

\$result:=SR Set Area(eReportArea;pSR_Report)

End if

End case

At this point, we are ready to begin using the SuperReport Pro editor. However, before we proceed, let’s add the code necessary for saving your report information to our report data field so that we can access it again.

Step 6: The last step we need to complete is saving our reports to the SuperReport Pro data field. Add the following code to your form’s accept button.

iSR_Err:=SR Get Area(eReportArea;pSR_Report) `copy data from plug-in are to variable

If (iSR_Err=0) `if the command completed successfully

[Report_Mgr]ReportData:=pSR_Report

End if

When you are retrieving information from a SuperReport Pro plug-in area, you must use a 4th Dimension BLOB variable, passing a BLOB field will not work correctly. Once the **SR Get Area** routine has completed successfully, you can copy the variable to your storage field as we have above.

Step 7: Now you are ready to begin using the SuperReport Pro editor. Enter the User Mode (or however you have configured access to the Report_Mgr table) and create a new record. You should see something similar to the following input form (this is the input form from the SuperReport Pro demo).

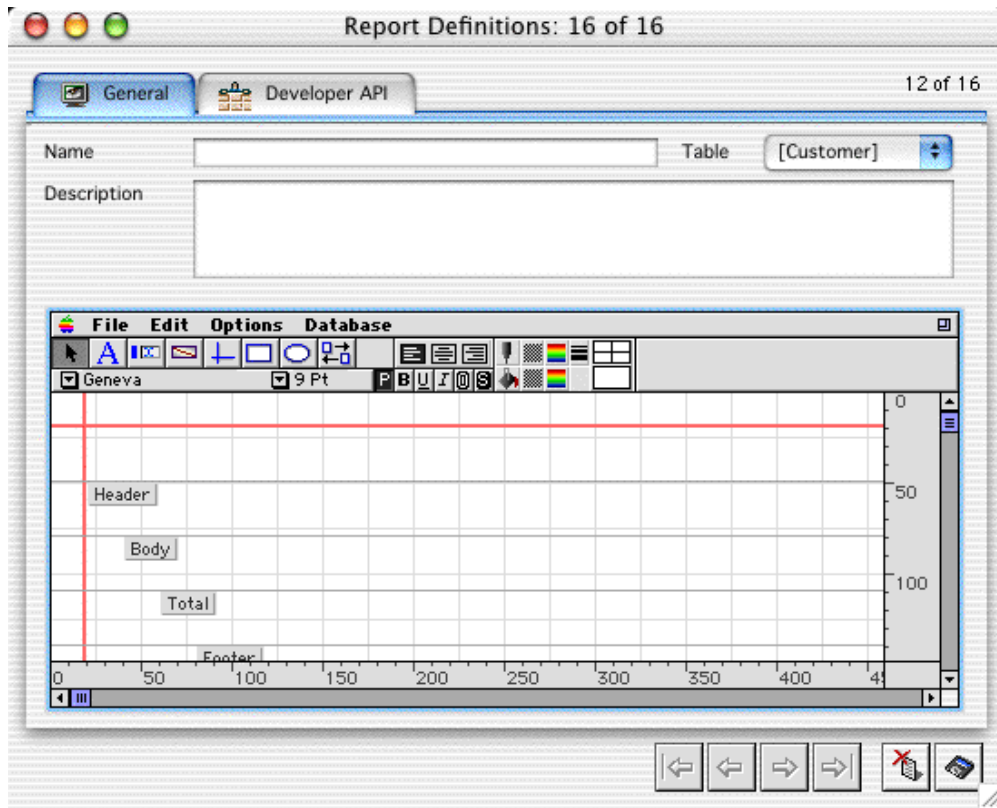


Figure 10 — SuperReport Pro on Input Form

Step 8: Go ahead and starting adding objects to the report form by using the Report Toolbar just to get a feel for what SuperReport Pro is all about. We'll go into more detail about the Report Editor later in this chapter.

Second Row



Figure 12 — Secondary Toolbar

Font Popup Tool — Assigns the selected font to the selected object(s).

Font Size Popup Tool — Assigns the selected font size to the selected object(s).

Font Style Group Tools — Assigns the selected font style to the selected object(s).

Foreground Pattern Popup Tool — Assigns a pattern foreground to the selected object(s).

Foreground Color Popup Tool — Assigns a color foreground to the selected object(s).

Section Marker Overview

SuperReport Pro provides the ability to create a variety of report types. When creating reports, you will need to adjust the Section markers (Figure 13) to support the style of reports you wish to create. There are four different section markers available when creating reports.

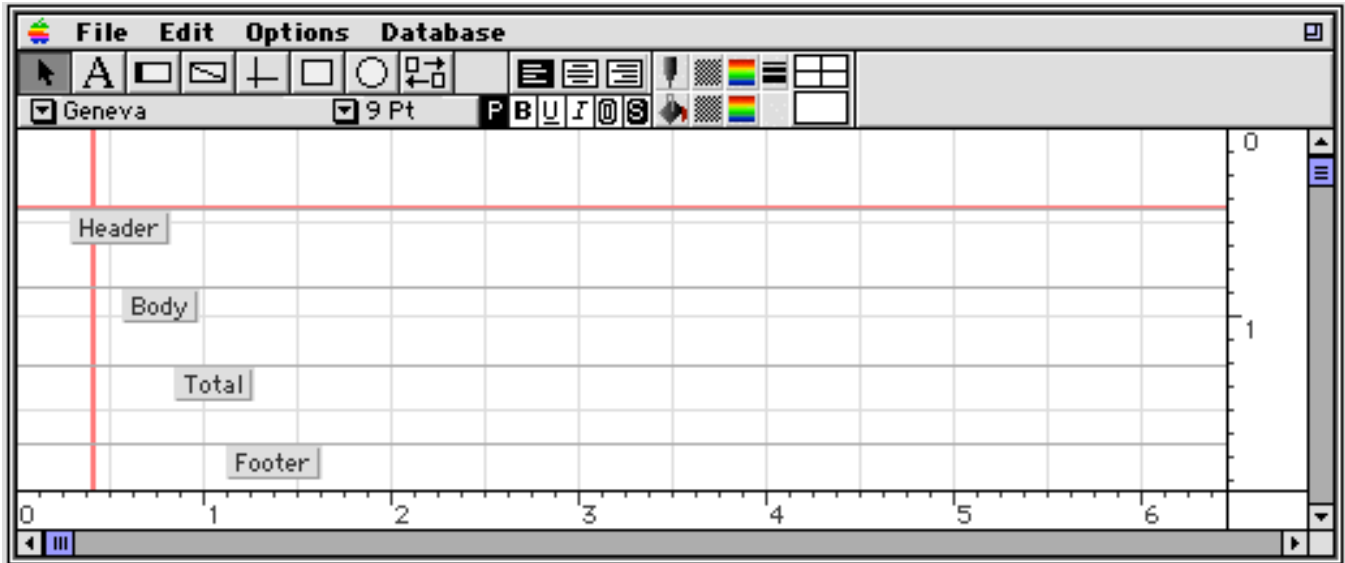


Figure 13 — SuperReport Pro Default Sections

- **Header**

The header section is printed once for each page. Typically, you will display report tiles, field titles, etc. in the Header section.

- **Body**

The body section is the primary section used by SuperReport Pro. This is where you will place the actual field or variable objects, which you wish to have appear on your reports. If you are creating a break-processing report (we cover this in a bit), you can also place field or variable objects in the additional break sections.

- **Total**

The total section is where you will place report totals. If you are creating break-processing reports, you can also use the SubTotal sections to provide total information.

- **Footer**

The footer section is printed once for each page. Typically, you will display report date and time information, or page numbers.

Menu Overview

SuperReport Pro provides a variety of menus and menu items, which can be used to further, customize your reports. For the sake of the Tutorial section, we'll introduce you to a few of the most commonly used menu items.

- **Edit | Modify Object (also displayed when you double click on a report object)**

This item will display the various object definition dialogs. Depending on the type of object you have selected, the corresponding dialog will be displayed.

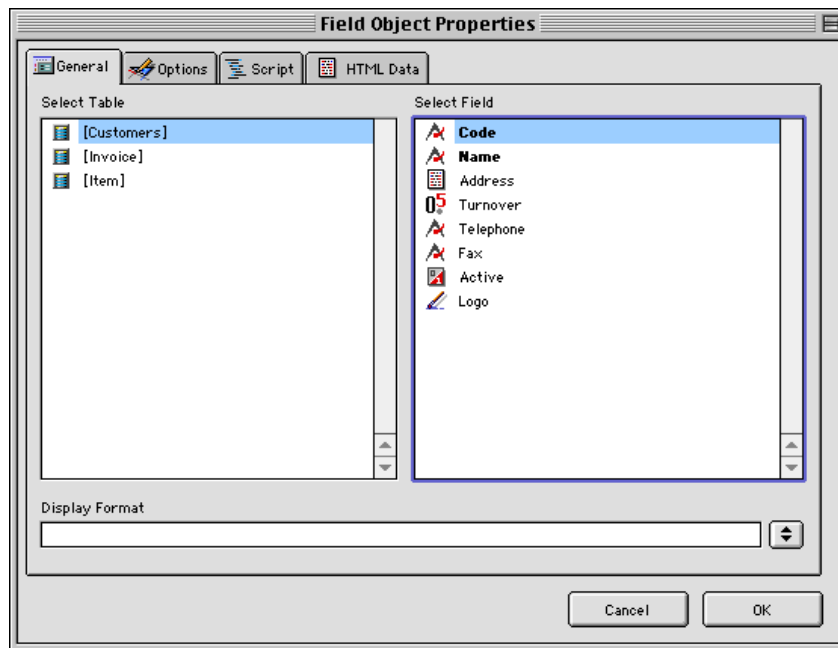


Figure 14 — SuperReport Pro Field Object Dialog

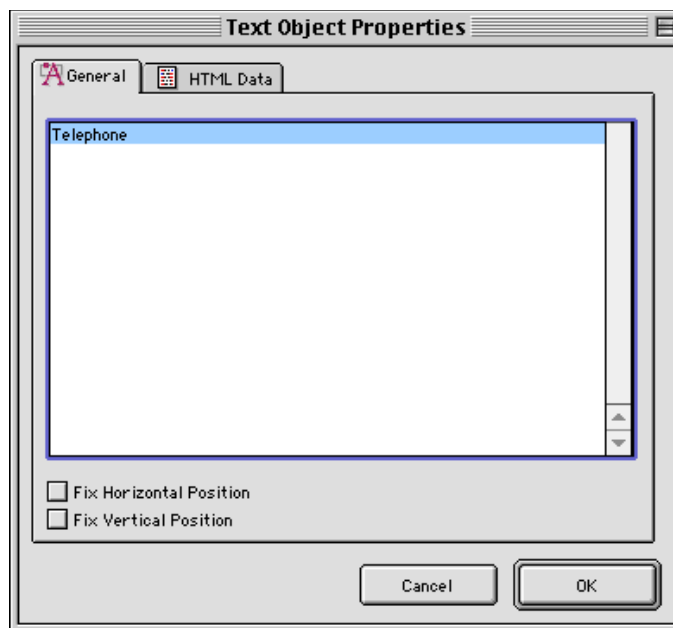


Figure 15 — SuperReport Pro Static Text Object Dialog

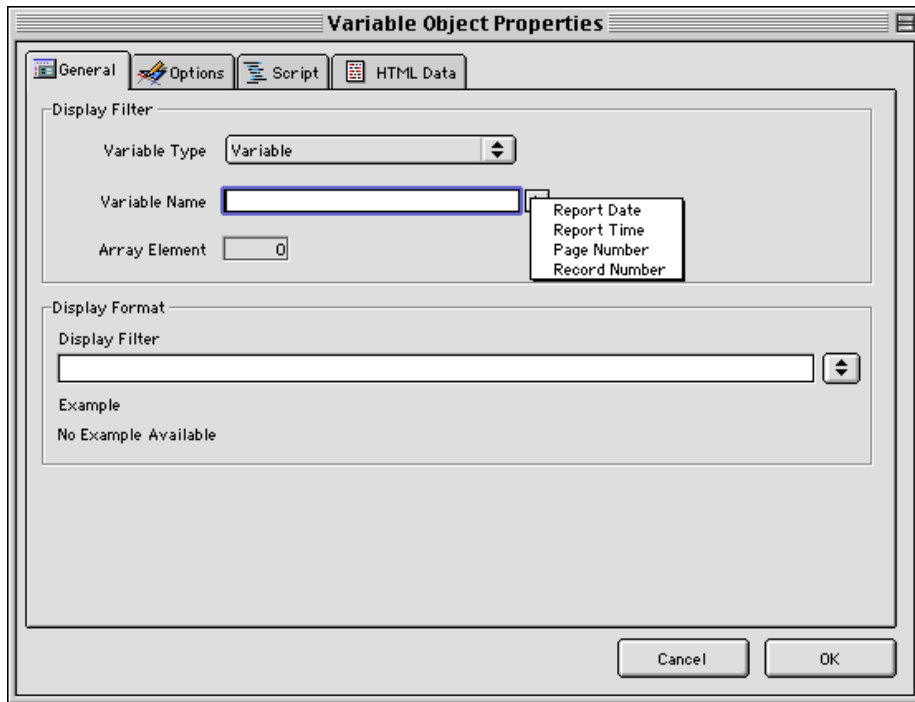


Figure 16 — SuperReport Pro Variable Object Dialog

- **Database | Report/Table Iterations**

This item will display the Report/Table Iterations definition dialog, which will allow you set the main file for your report.

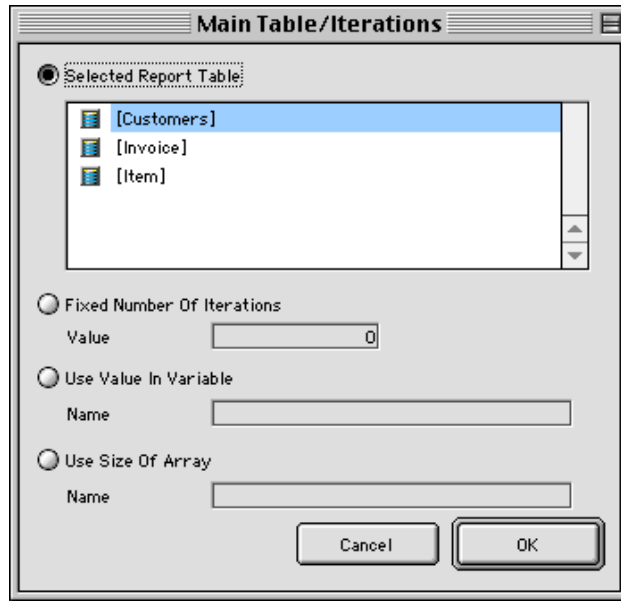


Figure 17 — SuperReport Pro Main Table / Iterations Dialog

- **Edit | Add/Delete Sections**

This item will display the Section management dialog, allowing you to add/delete report sections, which are used for creating break processing reports. You activate (or deactivate) sections by double clicking on the line for the appropriate section you wish to add/delete.

Note: Sections with bold attributes cannot be deactivated and will always be active.

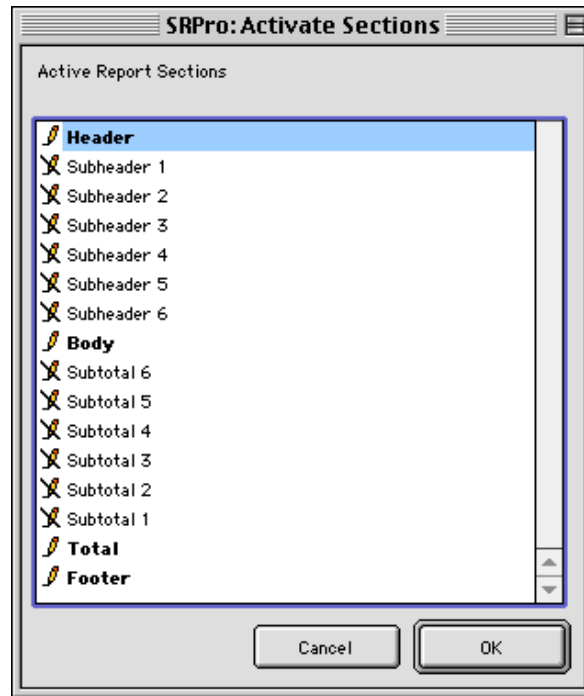


Figure 18 — SuperReport Pro Section Configuration Dialog

- **Database | Select Records**

This item will display the standard 4th Dimension Query Editor that will allow you locate the desired records, which you wish to print. If you don't perform any record selection, the current selection will always be printed.

- **File | Preview**

This item allows you to preview the report before it is printed. When displaying the report preview, you can move forward and backwards through the report (where applicable) and print the current page or entire report.

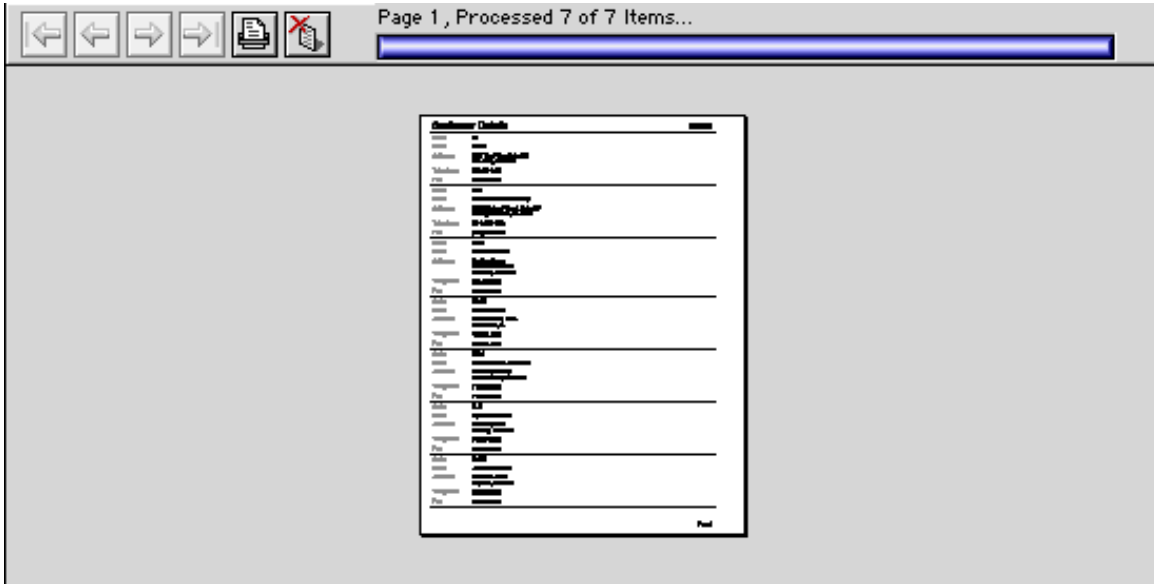


Figure 19 — SuperReport Pro Print Preview

Creating a basic listing report

Now that you have had an overview of the SuperReport Pro toolbar, section markers, and basic menus, you are ready to begin creating your first report. This example will guide you through the steps in creating a basic listing report.

Step 1: First, let's assign some information to the [Report_Mgr] fields.

Name (name of our report):	Customer Listing
Description (brief description of report)	Basic listing report
For File (popup menu)	[Customer]

Step 2: Now, add a new field to the report area. Each of the items should be placed in the Body section (above the Body marker). Select the Field Tool, then draw a box in the location where you would like the field located.

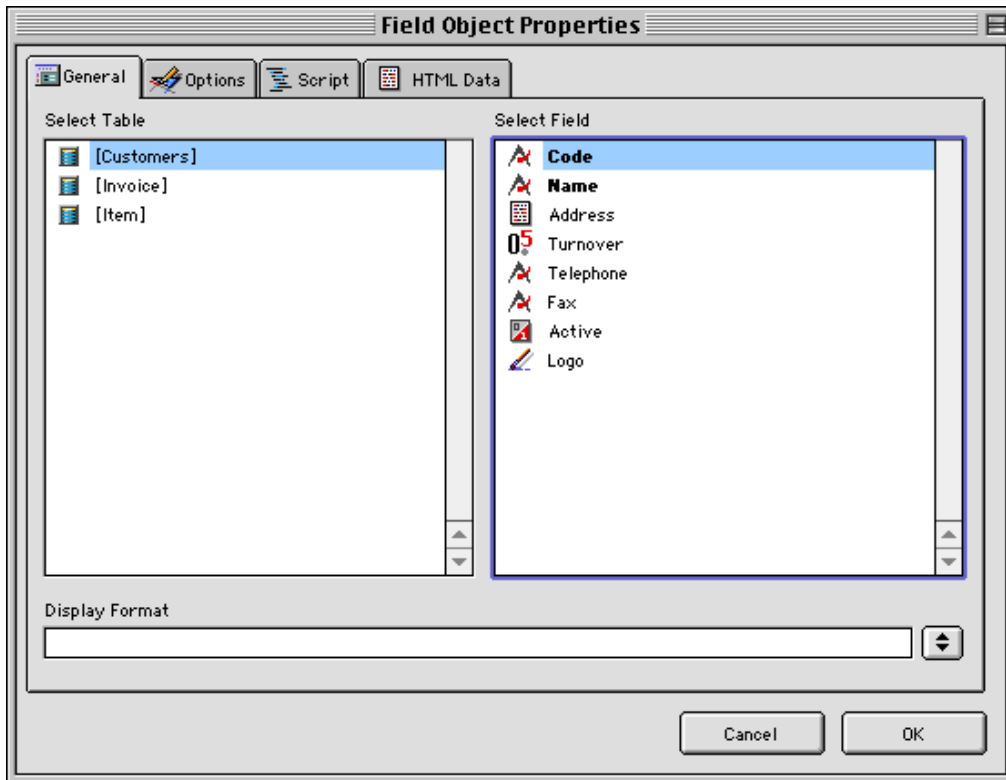


Figure 20 — SuperReport Pro Field Selector Dialog

The Field Definition dialog will be displayed. From the Field popup menu, select the field that you would like to place on the report (select Name) for the new object.

Step 3: Repeat Step 2 to add a couple more fields (add Address and Telephone). Your report should look something like the following:

Step 4: Now, we'll add the corresponding field titles. Using the Text Tool, create field titles for each of the fields we have added in Steps 2 and 3. While you are here, go ahead and apply some addition font formatting, such as making the titles bold. After you have completed adding the titles, your report should look something like the following:

Step 5: The next step is to set the Main Table for this report. The main file will be used table, which contains the current selection of records, in our case we'll select the Customer table. Select **Report Table/Iterations** from the **Database** menu. The Report/File Iterations dialog will be displayed.

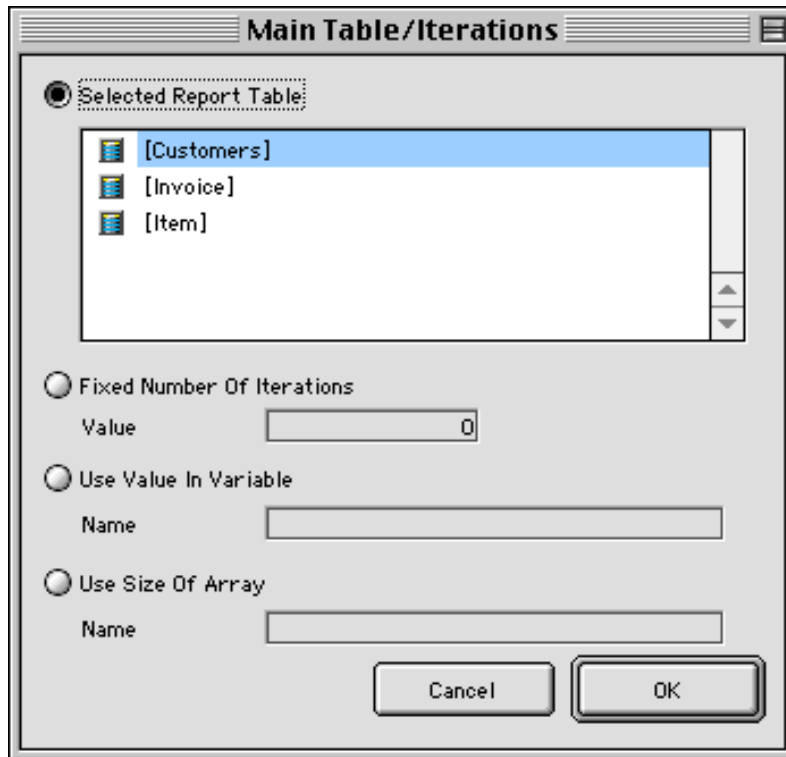


Figure 21 — SuperReport Pro Table Selector Dialog

- Step 6:** We're just about done. The last thing we need to do is establish a selection of records for the selected file. Choose Select Records from the Database menu. The standard 4th Dimension Query Editor will be displayed which will allow you to perform any query you wish to establish the record selection.
- Step 7:** You should now save your report before continuing. Click the forms Save button (which will in turn, execute the Save code we created in the **Using Plug-In Area** section).
- Step 8:** Return to the record we just created by double clicking it from the report table output form. If you have successfully completed the steps outlined in the **Using Plug-In Area** section, your report will appear in the report editor and you are ready to Print your Report!
- If you don't see the report template we just created, please review the Using Plug-In Area section and make sure you have completed the related steps.
- Step 9:** Select Print or Print Preview from the File menu to print your report. If you don't see any data, you will need to perform Step 6 again to establish a selection. SuperReport Pro does not maintain the selection, you will need to perform this task yourself before you print your reports.

Creating a basic form report

Congratulations, you have successfully created your first report! Now we'll create a little more complex report, a form report. Like the listing report we just created, you'll make use of the SuperReport Pro section markers. In addition, we'll introduce a new dialog, the Section Definition dialog.

Using the Section Definition dialog, you can customize the way SuperReport Pro prints your report. In the case of a form report, we don't want the records to print one after another as they were in the listing report. Instead, we want to have each record appear on it's own page.

The following steps will outline how to create a form report so that each record is printed on a separate page. Since you have already learned how to place objects on a report, we are going to skip that portion and concentrate on the specific of creating a form report.

Step 1: Using the SuperReport Pro demo, modify the "Customer Details" record. This contains a basic form report. In addition, you are welcome to create your own report, moving the body section down as necessary to create a form report.

TIP: You can hold down the shift key while moving a SuperReport Pro section marker, all section markers under the selected marker will move accordingly.

Step 2: Double Click on the Body section marker, display the Section Definition dialog.

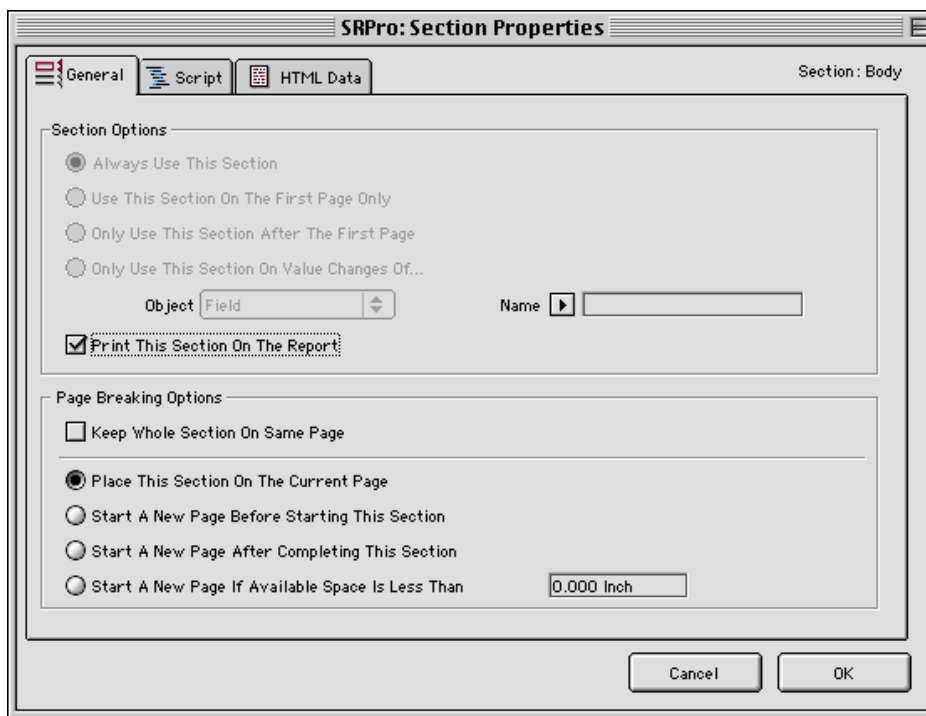


Figure 22 — SuperReport Pro Section Dialog (Body)

Step 3: Select the "Start A New Page Before Starting This Section" radio button in the Page Breaking Options section.

Step 4: Now select the Print Report or Print Preview menu item from the File menu to see your new form report. As you can see, each record is printed on a separate page.

Using Repeating Objects

One of SuperReport Pro's unique features is its ability to printed related many records as repeating objects. Using repeating objects, you create your reports in the same fashion, as you normally would, place parent table information in the body section.

With most reporting tools (such as the 4th Dimension form editor), if you want to print related many records, you are required to convert your reports to be based on the Many table, then print related one records.

SuperReport Pro provides the ability to include related many records as part of the primary table.

NOTE: You can only expand one level, if you need to print related many records more than one level, you'll need to redesign the report to print from the Many table.

Step 1: Create the report as you normally would, using the fields from the parent table, etc.

Step 2: Next, we'll place the following fields in the Body section, fields from a related many table.

[Invoice]CreateDate
[Invoice]Total

You can apply any type of data formatting you wish to get a feel for what SuperReport Pro has to offer in the way of displaying data.

Step 3: Double click on the [Invoice]CreateDate field and click the **Repeat** button in the Object Definition dialog (Options page), the following dialog will be displayed.

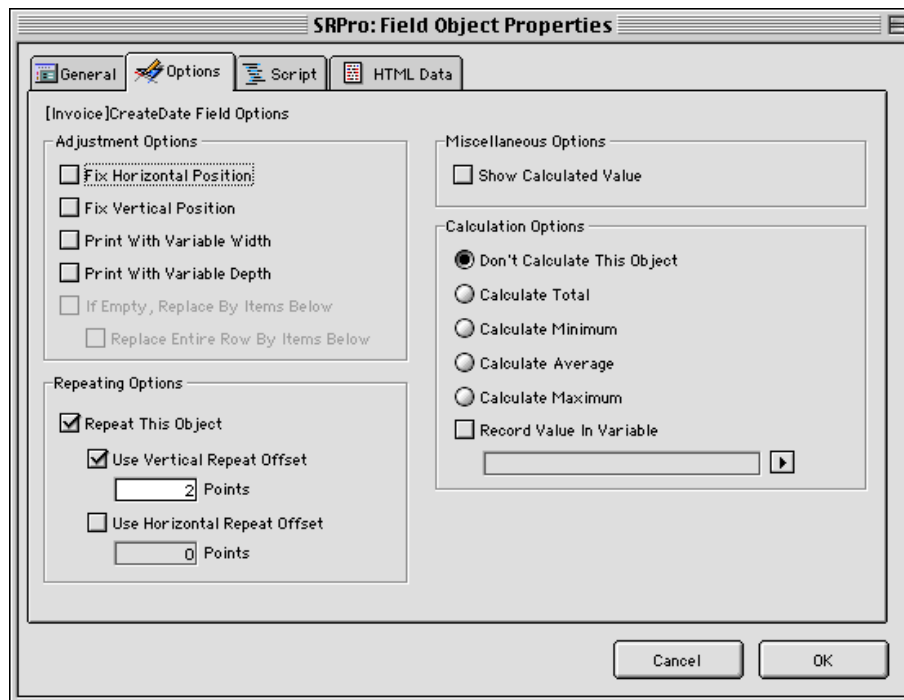


Figure 23 — SuperReport Pro Object Properties (Options tab)

Step 4: Select the Repeat This Object check box and click the Use Vertical Repeat Offset checkbox, assigning a value of 2 points. This value will be the number of pixels between each repeated object.

Step 5: The next step is optional depending on type of table relationship you have setup for the parent and many files.

- If your file structure is configured to use Automatic Relations, you don't need to perform any further tasks.
- If your file structure is configured to use Manual Relations, you will need to add the following code in the Body section markers scripts so that SuperReport Pro can get the related records at print time.

RELATE MANY([Customer])

Step 6: Save your report, then return to our new report.

Step 7: Now select the Print Report or Print Preview menu item from the File menu to see your new form report. As you can see, each record is printed on a separate page.

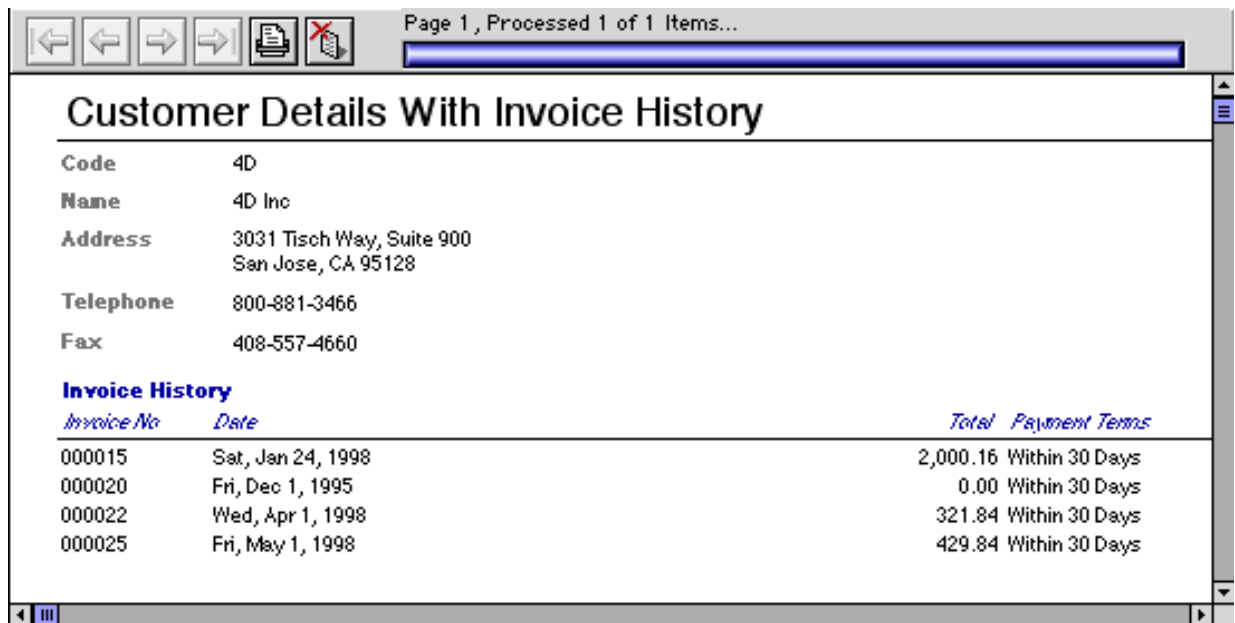


Figure 24 — SuperReport Pro Repeating Example Preview

If you would like to review this report, use the SuperReport Pro Demo and open the “Customer Details With Inventory History” sample record.

Creating a break-processing report

SuperReport Pro provides a very powerful break processing reporting system, providing up to six (6) break levels. Using a break processing, you could create a report that allows you to print a list of invoices, sorted by customer.

Step 1: The first step will be to define the Main File for our report. **Select Report/File Iterations** from the **Database** menu and select Invoice

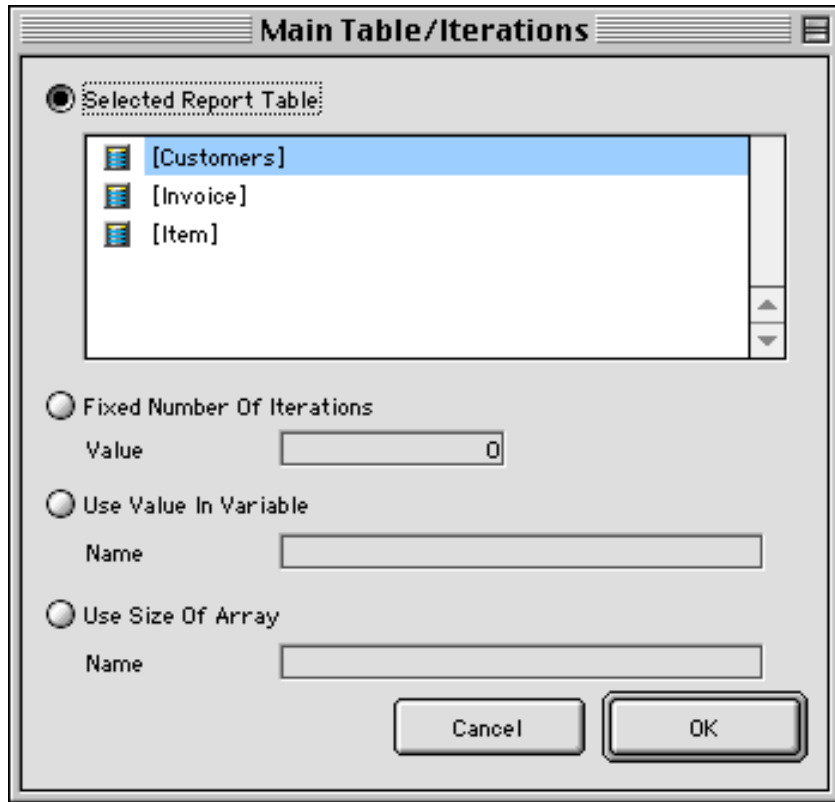


Figure 25 — SuperReport Pro Main Table/Iterations Dialog

Step 2: The next step will be to setup the sections so that we are going to be using on this report. Select Add/Delete Section from the Edit menu. When the Section Definition dialog is displayed, double-click the SubHeader 1 and SubTotal 1 items.

Note: Sections with bold attributes cannot be deactivated and will always be active.

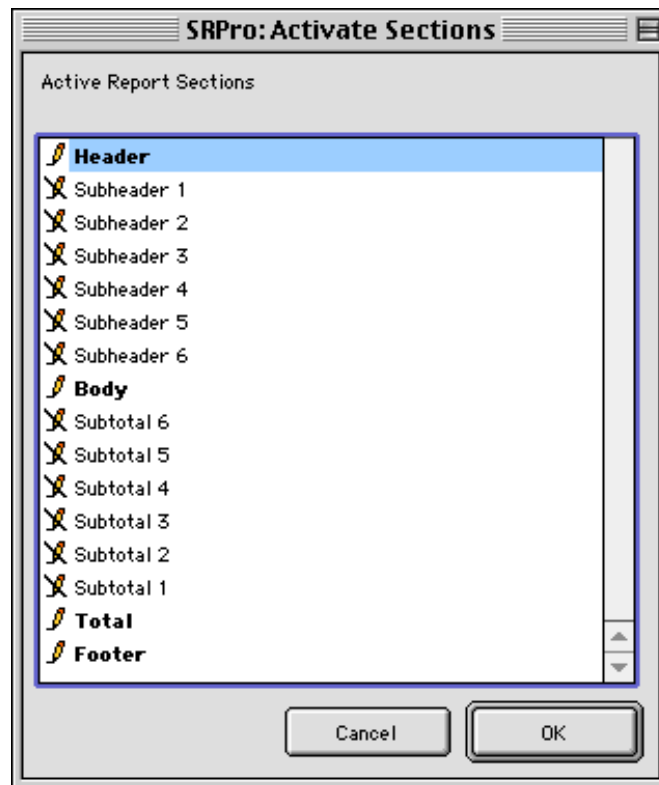


Figure 26 — SuperReport Pro Section Dialog

Step 4: The next step will be to configure the “Body” section script to set the vTotalTxt variable located in the SubTotal 1 section. Double click on the “Body” section marker to display the Section Definition dialog and click the “Script...” tab.

Enter the following text, then click the OK button to accept the script:

```
vTotalTxt:="Total for Period "+[Invoice]Period+" "
```

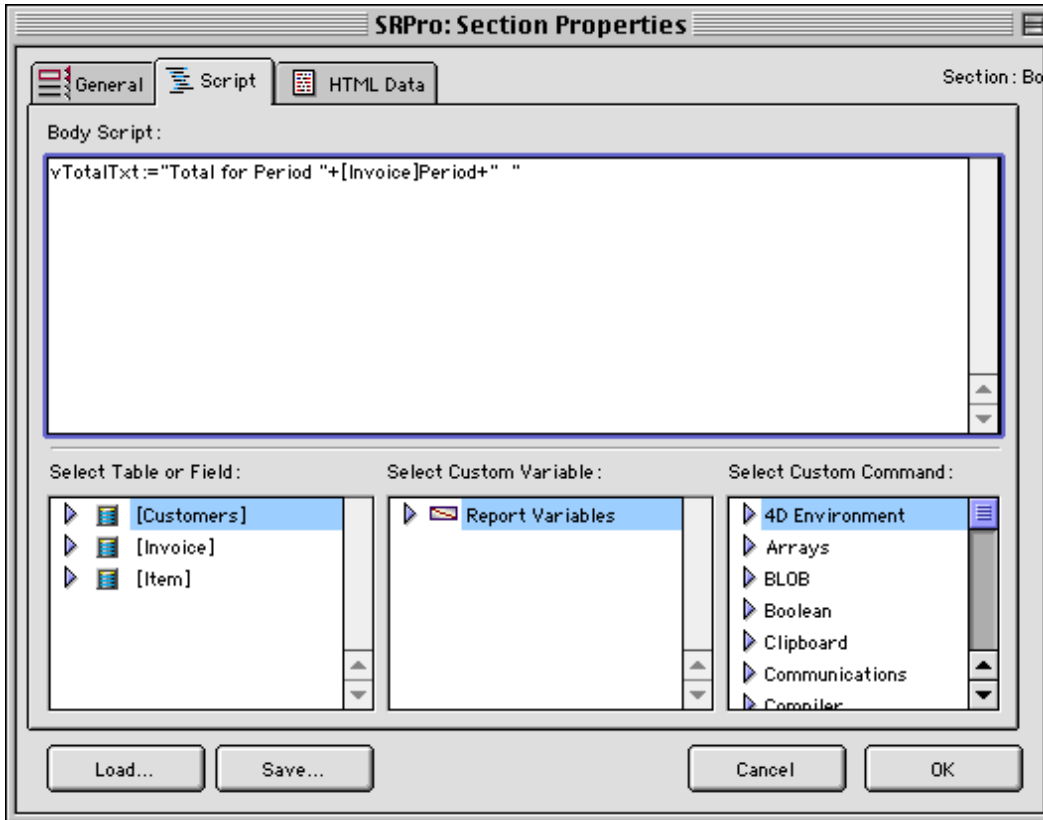


Figure 27 — SuperReport Pro Section Dialog

Step 5: Next, we need to configure the SubHeader 1 section so that our break processing occurs correctly. SuperReport Pro calculates the breaks in a report based on one of three items.

- The value of a field for a given file
- The value of a variable
- The value of an array element

In our example, we are going to use the value of a field, the `[Invoice]Period` field.

Step 6: Double click on the SubHeader 1 section to display the **Section Definition** dialog. You will notice the **Only use this Section on Value Change Of...** item is enabled (page 1).

- Select Only use this Section on Value Change Of...
- Select Field from the Object popup menu
- Select [Invoice]Period from the field popup menu

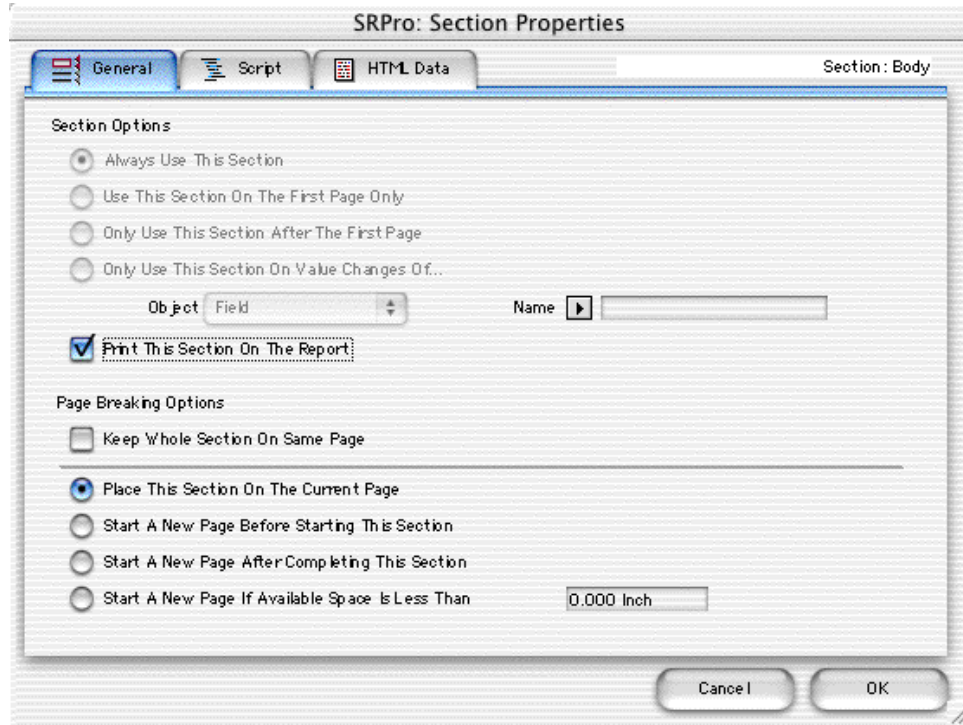


Figure 28 — SuperReport Pro Subheader Dialog

Step 7: The next step to making our making our break processing report work correctly is to establish a selection of records and define a sort order. If you don't sort your selection, SuperReport Pro will not be able to correctly calculate the break, thus your report will not appear as you would expect.

Step 8: Select **Start Script...** from the **Scripts** submenu, located in the **Database** menu. The SuperReport Pro Script Editor will be displayed. We'll be providing more detail about SuperReport Pro scripts in the next section

Enter the following code:

```
ALL RECORDS([Invoice]) `print the selection of invoice records  
ORDER BY([Invoice];[Invoice]Number;>)
```

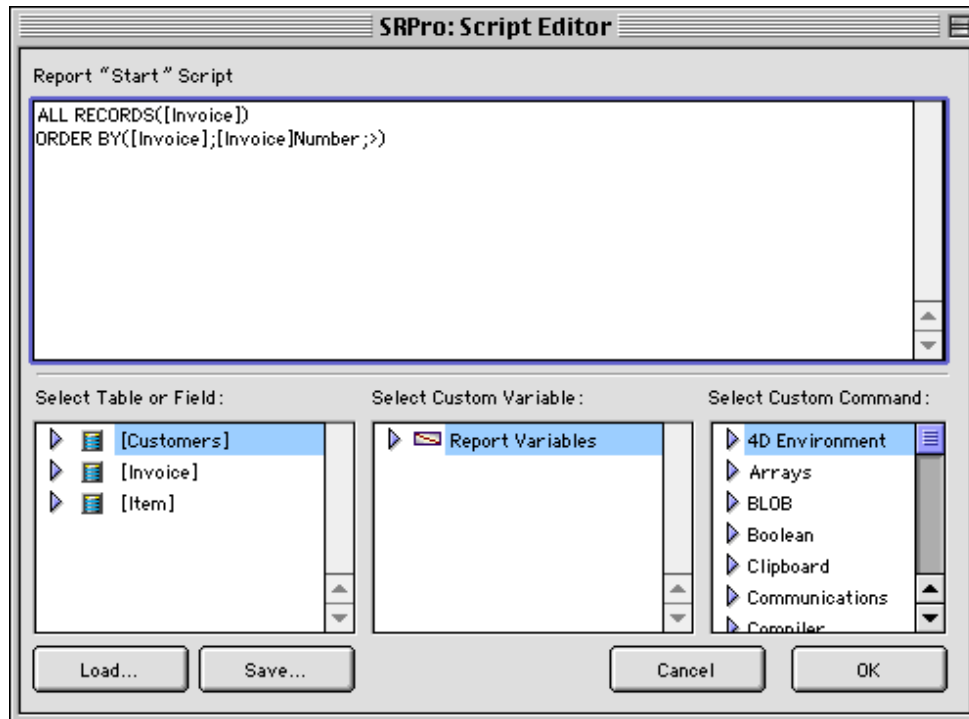


Figure 29 — SuperReport Start Script Dialog

Step 9: Save your report, we have done a lot of work!

Step 10: Open the record we just created and select Print Report or Print Preview menu item from the File menu to see our report. Your report should look something like this (we selected Print Preview)

<i>Invoice No</i>	<i>Date</i>	<i>Total</i>	<i>Payment Terms</i>
000004	Fri, Jan 1, 1999	2,700.00	Within 30 Days
		<i>Total for Period 9901</i>	2,700.00
<i>Invoice No</i>	<i>Date</i>	<i>Total</i>	<i>Payment Terms</i>
000005	Wed, Feb 1, 1995	1,723.68	Within 30 Days
000009	Sun, Feb 5, 1995	4,772.52	Within 30 Days
		<i>Total for Period 9502</i>	6,496.20
<i>Invoice No</i>	<i>Date</i>	<i>Total</i>	<i>Payment Terms</i>
000010	Sat, Jan 30, 1999	2,921.11	Within 4 Days
		<i>Total for Period 9901</i>	2,921.11
<i>Invoice No</i>	<i>Date</i>	<i>Total</i>	<i>Payment Terms</i>
000011	Fri, Mar 3, 1995	1,240.92	Within 30 Days

Figure 30 — SuperReport Preview Sample

Creating an HTML Report

SuperReport Pro enables users and developers the ability to include HTML tags in report objects which will be utilized when printing a report using SR Print HTML or by selecting HTML option when selecting Print To Disk from within the Report Editor.

Each object may contain a standard tag, which is sent before the actual data of the object, and an end tag, which is sent after the object data.

Furthermore, when reports are printed in HTML format, all font attributes are retained and the corresponding font tags are sent automatically when the report is created.

The following steps will convert an existing report (the Basic Listing Report created in the first example) to be suitable for HTML output.

Step 1: Select **Database | Scripts | Start Script** and enter the following code. This will establish some variables which are used within the HTML report, as well as establish a record selection to print.

```
tBegTagHTML:="<html><body bgcolor='#ffffff'>"  
tEndTagHTML:="</body></html>"
```

```
ALL RECORDS([Customer])  
ORDER BY([Customer];[Customer]Code;>)
```

Step 2: The next step will be to configure the Header section, which is where you assign the standard HTML header information. In addition to standard HTML tags, we'll introduce the feature of using 4th Dimension variables as HTML token variables.

Double click on the Header object, select the HTML Data button, and enter the following text:

```
<%tBegTagHTML%>  
<table border=1 width=100%>
```

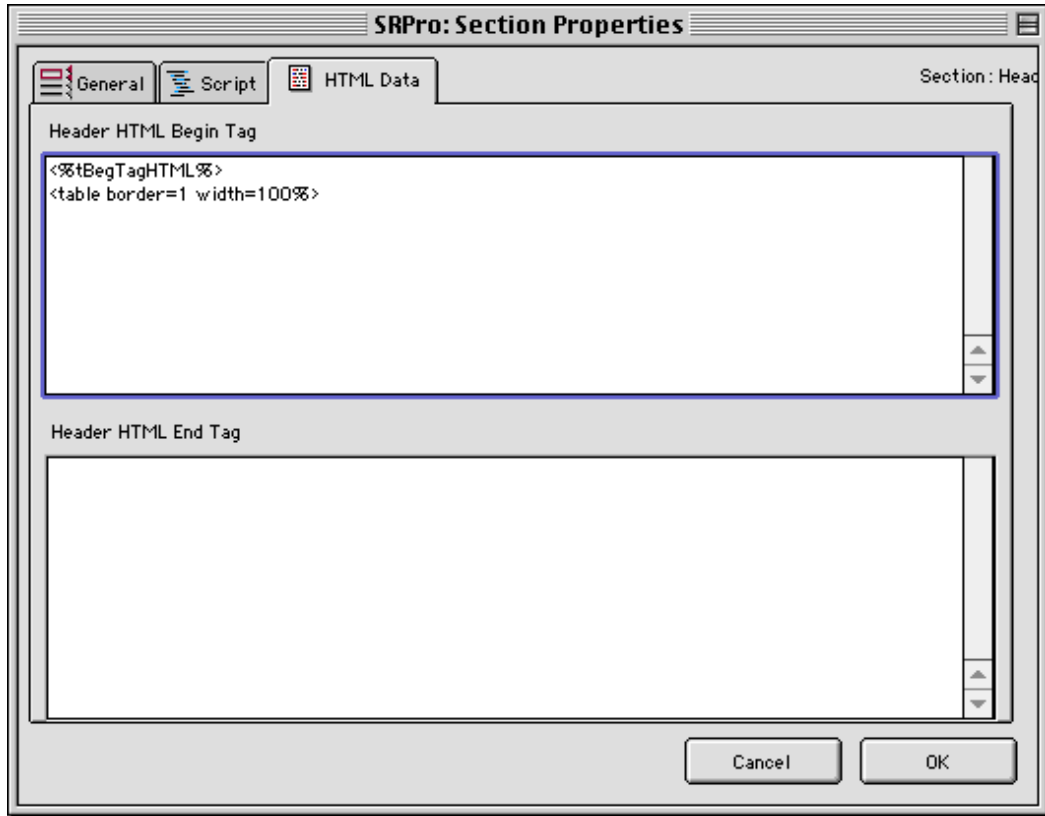


Figure 31 — SuperReport Pro Header Section | HTML Data

The `<%tBegTagHTML%>` section tells SuperReport Pro to insert the contents of the variable “tBegTagHTML” when this report is printed in HTML format.

The next line will create a standard HTML table to display our data. . Since we are creating a listing style report, using an HTML table is a logical choice.

Step 3: Since we have already created the table creation tag, we'll jump forward a bit to the "Total" section where we placing the corresponding `</table>` tag. We place the close table tag in this section as it is where the report will actually when the corresponding section completes printing.

Double click on the Total section marker, click on the HTML Data button, and enter the following HTML tag in the Before section:

`</table>`

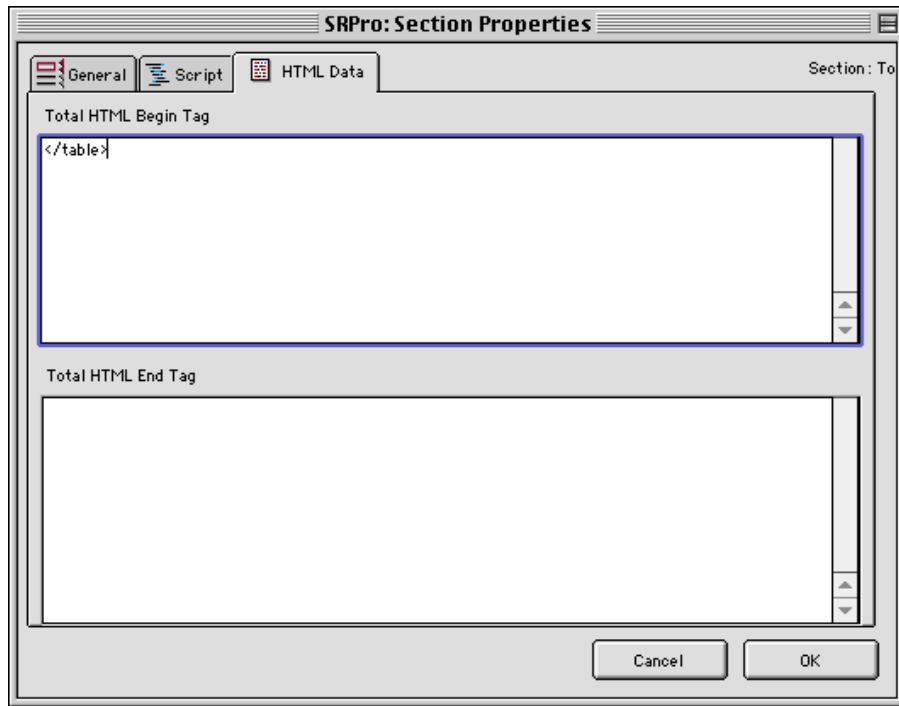


Figure 32 — SuperReport Pro Total Section | HTML Data

If you were to create a break processing report where you had multiple sections, you could create nested tables by place the `<table>` tag in the SubHeader section, and the `</table>` tag in the corresponding SubTotal section.

Step 4: The next step will be to insert the new row tags (<tr> and </tr>). Since the Body script is executed for each record in the current selection and it executes before the body sections object scripts (for more information on the SuperReport Pro Execution Cycle, please refer to **Chapter 4 - SuperReport Pro Fundamentals (Understanding SuperReport Pro Execution Cycle)**).

Double click on the Body section marker, click on the HTML Data button, and enter the following HTML tags.

Start Tag: <tr>
End Tag: </tr>

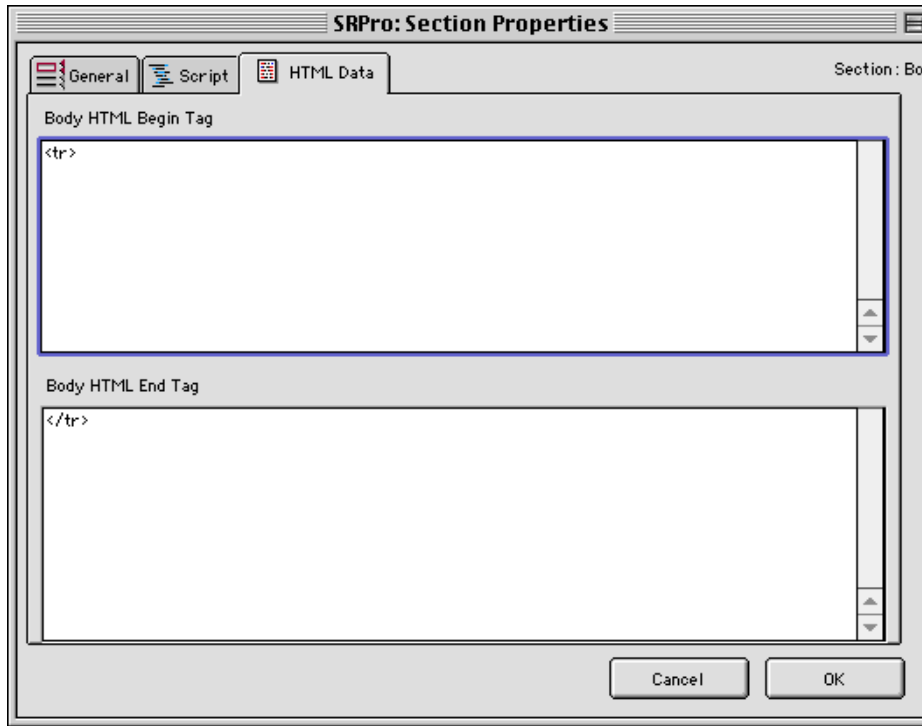


Figure 33 — SuperReport Pro Body Section | HTML Data

Step 5: Now that we have successfully created our tables, it is now time to add some data cells. Using the HTML `<td>` and `</td>` tags, we can create table cells for the current row.

For each field in the body section, double click on the field object, click on the HTML button and add the following tags:

Start Tag: `<td>`
End Tag: `</td>`

As each object in the body section is printed, a new cell will be created.

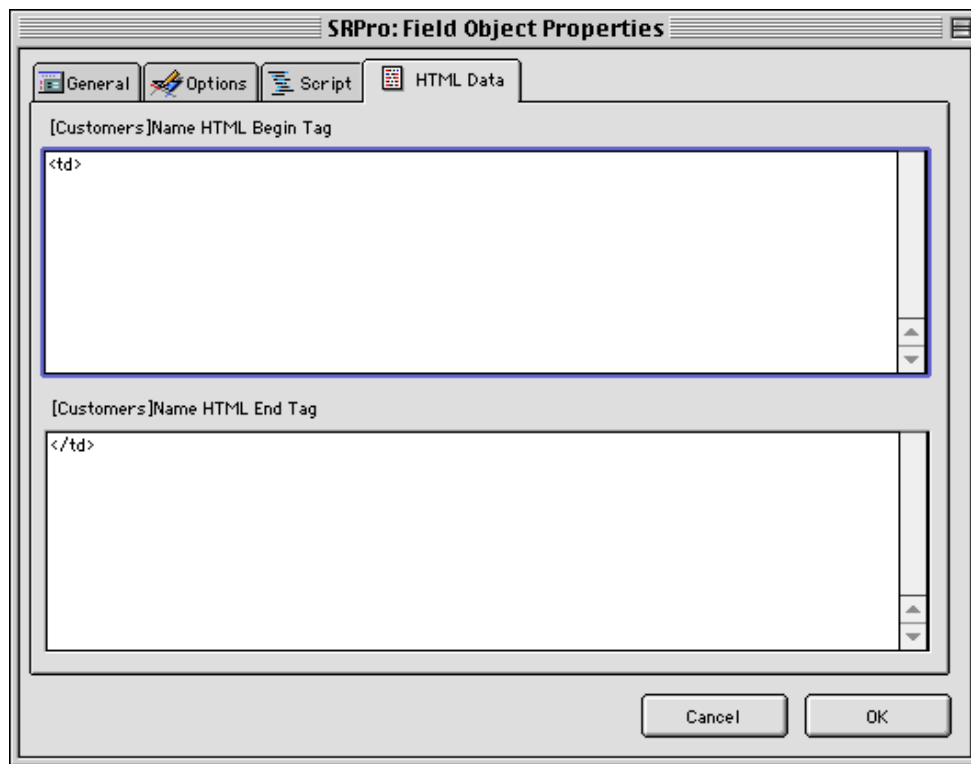


Figure 34 — SuperReport Pro Object | HTML Data

Step 6: The last piece of the body data will be to create a row for the data blocks in the header section. Since the scripts of these objects are executed after the header marker, we can create the first row of column data before the corresponding body section is executed.

For each object in the header section, which is a column title, double click on the object, select the HTML button, and add the following tags:

Start Tag: <td>
End Tag: </td>

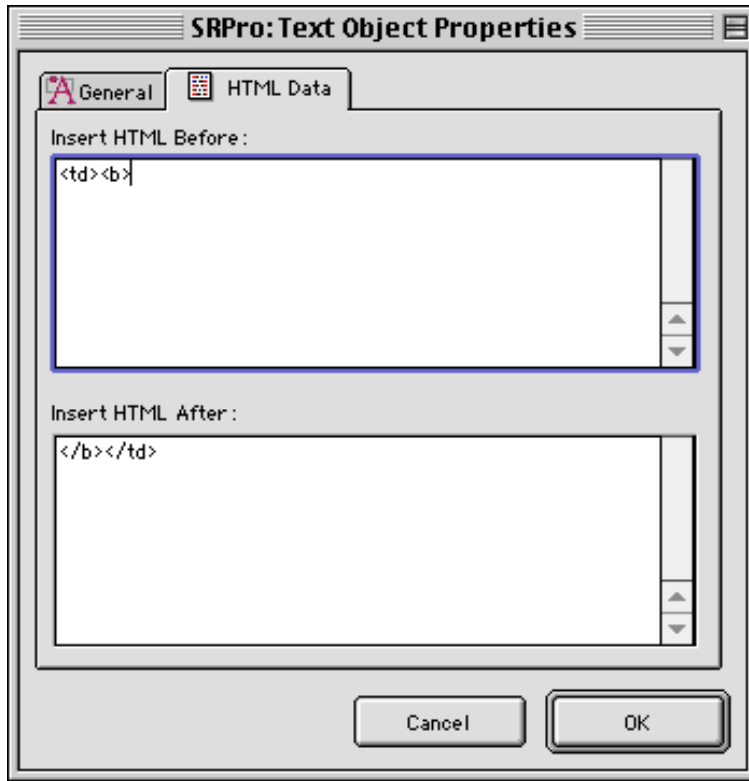


Figure 35 — SuperReport Pro Header Object | HTML Data

These tags will create a single row containing the column headings, applying the bold HTML tag. You will notice that End Tag section has the tag before the </td> tag. As outlined in the first Step, the Start Tag is sent to the HTML file before the data is sent, the data is sent, and then the end tag is sent. If you were to place the tag after the </td> tag, the generated HTML file would be incorrect.

Step 7: The final configuration item will be to close our HTML file. Since we placed the beginning HTML tags in the reports Header section marker, the logical place to place the closing HTML tags is in the Footer section.

Double click on the Footer section marker, click on the HTML button, and enter the following code.

```
<%tEndTagHTML%>
```

Like the HTML header, the end tag is using a custom 4th Dimension variable to hold the standard end tag data.

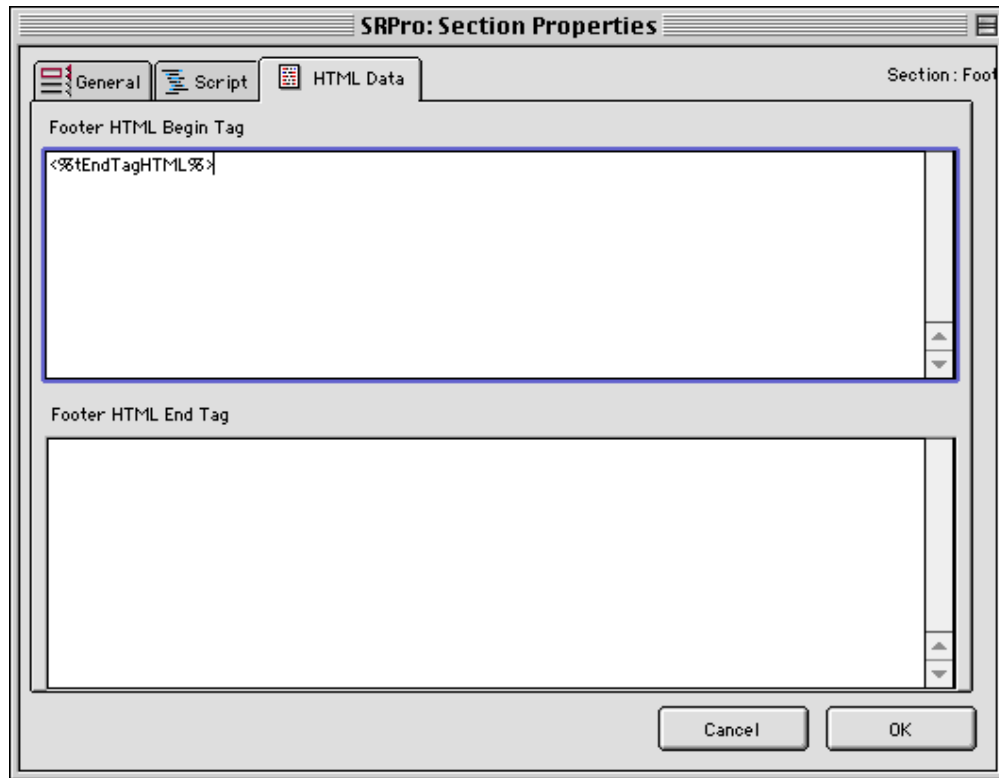


Figure 36 — SuperReport Pro Footer | HTML Data

Step 8: Now we are ready to create our HTML file, using the **SR Print HTML** command or by selecting **Print To Disk...** from the **File** menu, selecting the **Output As HTML** option in the **Output Options** section.

When printing to disk using HTML output, remember to select the Static Text items and all sections, otherwise your HTML data will be incomplete.

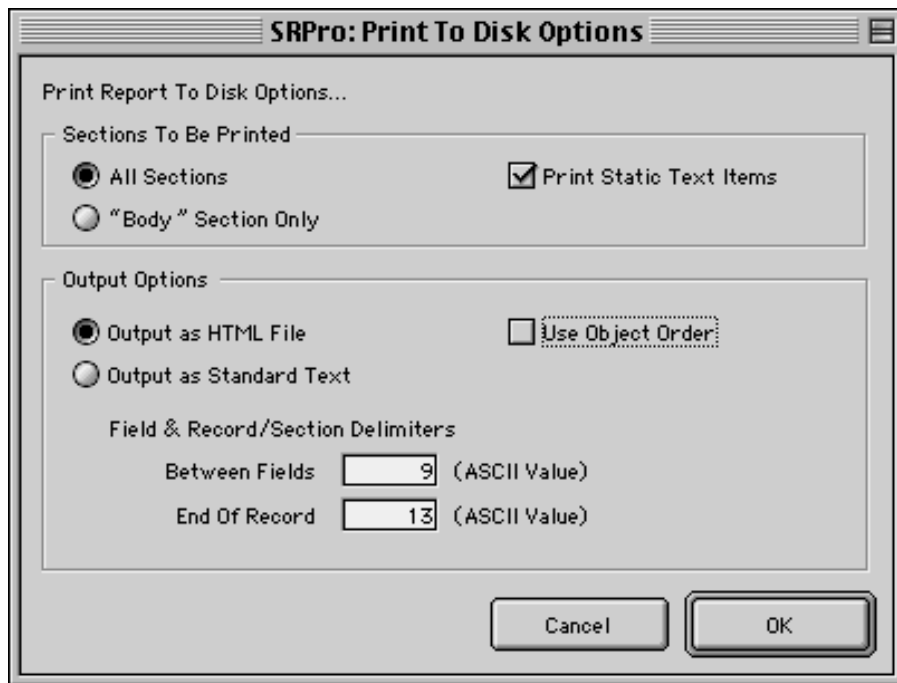


Figure 37 — SuperReport Pro HTML Print

Printing Arrays

While SuperReport Pro provides an extensive feature set for printing 4th Dimension database fields, we didn't stop there! SuperReport Pro can also print 4th Dimension arrays, whether it be a full array (all elements in the array), or a specific array element — SuperReport Pro can handle it!

The following tutorial will teach you how to create a SuperReport Pro report for printing a group of arrays. In addition, we'll discuss how to configure the Report Table/Iterations dialog to use the array size instead of the number of records in a selection as we have in previous examples.

Using the Customer Listing report we created in the first example as a template, perform the following tasks.

Step 1: Replace all field reference objects with new variable objects.

Click on the variable tool in the SuperReport Pro Editor Toolbar, and draw a box in the same location as the original [Customer]Code field, and assign a variable name of **aCCCode** (for Customer Code) and select the Variable Type to be Array (Automatic).

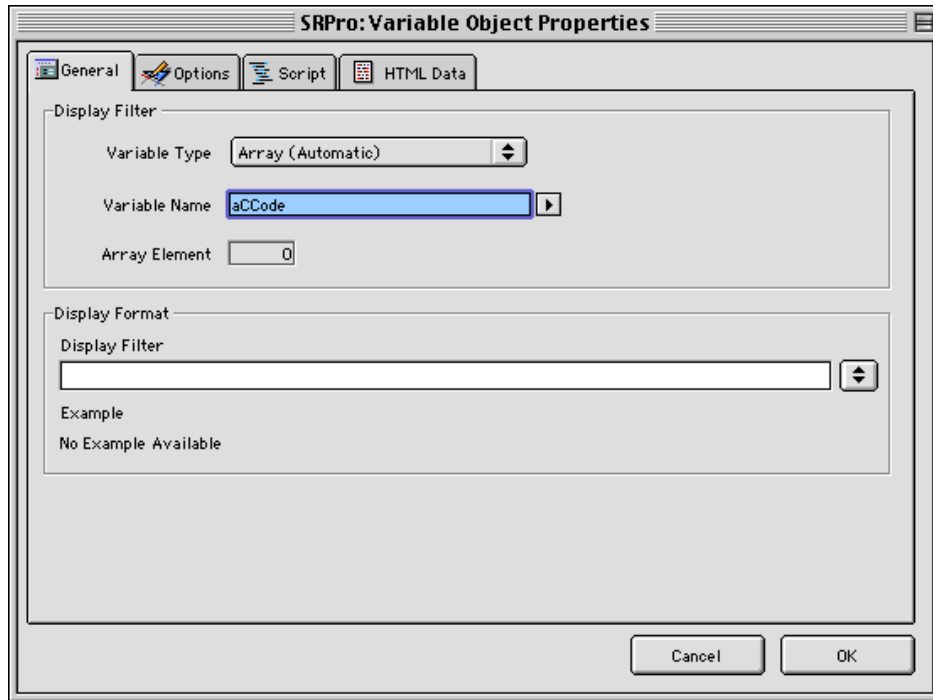


Figure 38 — Array Variable Object

Step 2: Repeat **Step 1** for each of the following variable names

aCName	Customer Name
aCTel	Customer Phone

When you have completed Step 2, your report should look simple like this:

Customer List				Report Date
Code	Name	Telephone	Fax	
{aCCode}	{aCName}	{aCTel}	{aCFax}	
Body				
Total				Page Number
Footer				

Figure 39 — Sample Array Report

Step 3: The next step is to define the report iterations based on the size of the array we are printing. Since all the arrays will be equal size, we can select any of the array's we wish to define the report iterations.

Select **Report Table/Iterations** from the **Database** menu. When the dialog is displayed, select the **Use Size Of Array** radio button and use the **aCCode** array.

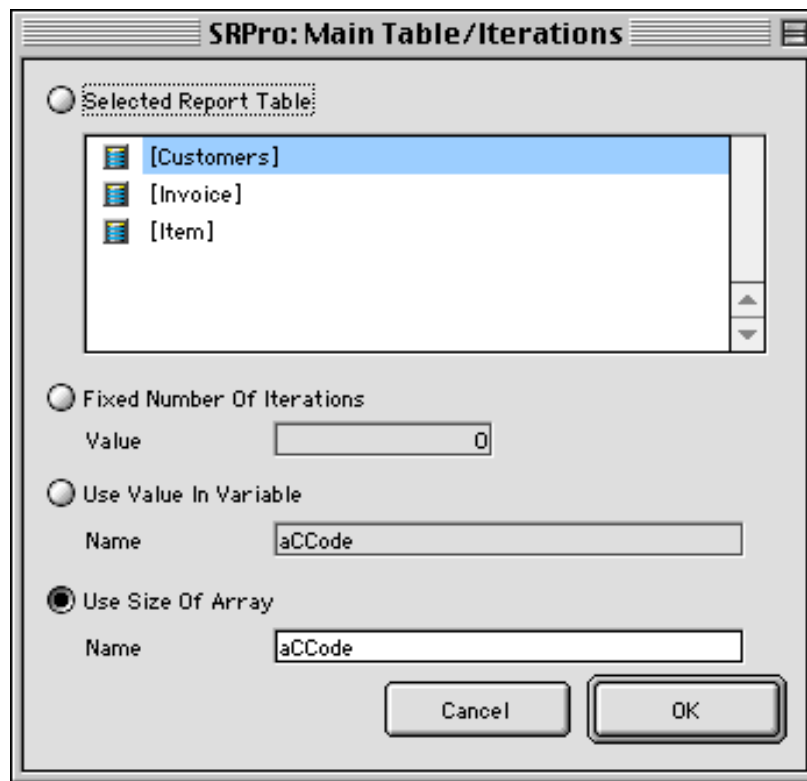


Figure 40 — Report Iterations based on array size

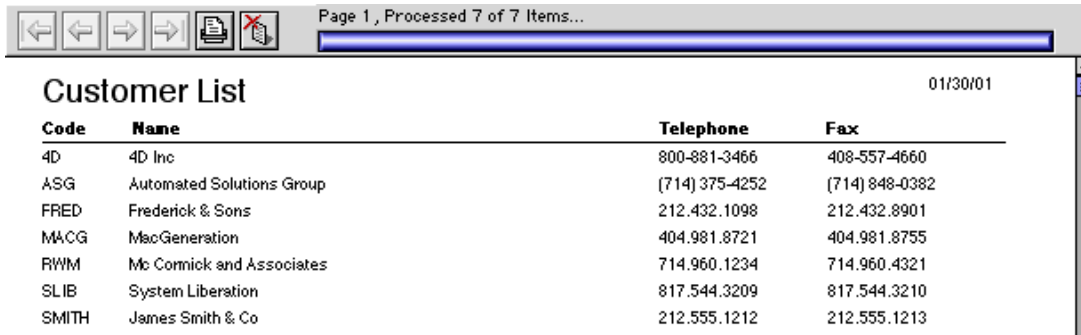
Step 4: The last step is to create the arrays. Select Start Script from the Scripts submenu, located in the Database menu.

Enter the following code and save the script:

```
ALL RECORDS([Customer])
SELECTION TO ARRAY([Customer]Code;aCCode;[Customer]Name;aCName;[Customer]Telephone;aCTel)
SORT ARRAY(aCCode;aCName;aCTel;>)
```

Step 5: Again, we'll save our report, then re-open the record.

Step 6: Select **Print Preview** or **Print Report** from the **File** menu to see our work!



The screenshot shows a software interface for a report preview. At the top, there is a navigation bar with icons for back, forward, and search, and a status bar indicating "Page 1, Processed 7 of 7 Items...". The main content area displays a table titled "Customer List" with a date stamp "01/30/01". The table has four columns: Code, Name, Telephone, and Fax. The data rows are as follows:

Code	Name	Telephone	Fax
4D	4D Inc	800-881-3466	408-557-4660
ASG	Automated Solutions Group	(714) 375-4252	(714) 848-0382
FRED	Frederick & Sons	212.432.1098	212.432.8901
MACG	MacGeneration	404.981.8721	404.981.8755
RWM	Mr Cornick and Associates	714.960.1234	714.960.4321
SLIB	System Liberation	817.544.3209	817.544.3210
SMITH	James Smith & Co	212.555.1212	212.555.1213

Figure 41 — Print Preview of Array data

Step 5: The next step to creating our report will be to add a column (located in the body area), which will display actual database information.

The first section of code will define the beginning coordinates for the detail object. Remember, we have 25 pixels between sections, thus we need to move it down to compensate for the various section markers.

```
`set the base coordinates for the fields we are adding
$top:=60
$bottom:=$top+12
$SR_ObjType:=SR Object Type Field

`create the first column
$left:=50
$right:=$left+$objWidth
$tableNo:=Table(->[Customer]Code)
$fieldNo:=Field(->[Customer]Code)
$SR_ItemID:=SR Create Object($SR_AREA;$SR_ObjName;$top;$left;$bottom;$right;
    $SR_ObjType;$SR_Opts;$SR_Selected;$tableNo;$fieldNo;$SR_VarType;
    $SR_ArrayElement;$SR_CalcType;$SR_CalcName;$SR_Rows;$SR_Cols;
    $SR_HRepeat;$SR_VRepeat)
```

Step 6: After we have a column in the body section, we are ready to get some data. SuperReport Pro will use the current record selection for all reporting requests, unless you define a different selection of records.

Typically, SuperReport Pro developers will use the Database scripts to define the record selection. In this case, we'll use the "Start" script to define the record selection. In this case, we are going to display all records from the Customer table, sorted by [Customer]Name.

```
`set the selection
$SR_StartScript:="ALL RECORDS([Customer])"+Char(13)
$SR_StartScript:=$SR_StartScript+"ORDER BY([Customer];[Customer]Name;>)"

$SR_Err:=SR Set Scripts ($SR_AREA;$SR_StartScript;"";"")
```

Step 7: The final step before we save our report is to define the table, which will be used to generate the report selection. The following routine will set the "main table" (which is equivalent to selecting **Database | Main Table/Iterations** when using the SuperReport Pro Editor.

```
`set the main table for which the selection of records will be used
$SR_Err:=SR Main Table2 ($SR_AREA;SR MainTable Choose Table ;Table(->[Customer]);"")
```

Step 8: At this point, our report creation is complete and all that remains is to save the report to disk and destroy the offscreen area to release the memory occupied by the report object.

```
`save the report to disk
$SR_Err:=SR Save Report ($SR_AREA;"test.srp";0)
```

```
`delete the offscreen area
SR DELETE OFFSCREEN AREA ($SR_AREA)
```

Step 8: In the event you wish to "review" the report we just created, you can either open the form, which contains a SuperReport Pro area, or open the saved report from disk, or you can use the **Open External Window** command.

In this case, we'll load the report from disk and display it using an external window.

```
`open external window
```

```
$SR_AREA:=Open external window(50;50;Screen width-50;  
Screen height-50;8;"test.srp";"%SuperReport")  
If ($SR_AREA>0)  
  `load the report we just created  
  $SR_Err:=SR Load Report ($SR_AREA;"test.srp")  
End if
```

While this example demonstrates SuperReport Pro's ability to procedurally create reports, the control does not stop there. For a more complete report based on these examples, please refer to the X_CreateReport routine contained in the SuperReport Pro demo database.

This routine will demonstrate some more techniques you can use when creating reports, as well as a number of additional objects to create a "Real Life" report.

6 — Command Reference

This chapter outlines all the 4th Dimension procedure objects and supporting external routines (SuperReport Pro Extras) included in the SuperReport Pro package as well as examples on how to use the various SuperReport Pro routines.

The syntax used here is similar to the syntax used by 4th Dimension Language Reference manual and is broken down into the following sections:

- SR Pro – Access
- SR Pro – Consulting
- SR Pro – Menus
- SR Pro – Miscellaneous
- SR Pro – Printing
- SR Pro – Developer Interface (API)

Routine Overview

SuperReport Pro contains over 30 routines for creating and managing reports. Each of these routines falls into one of 5 categories for easy identification. SuperReport Pro uses the following categories:

- **SR Pro – Access**

Outlines the routines used to create and configure a SuperReport Pro report, using the SuperReport Pro plug-in area.

- **SR Pro – Customizing**

Outlines the routines used to customize the file structure and variables used by the SuperReport Pro plug-in area.

- **SR Pro – Menus**

Outlines the routines used to procedurally control the menus used by the SuperReport Pro plug-in area.

- **SR Pro – Miscellaneous**

Outlines the routines, which provide miscellaneous information and configuration options when using SuperReport Pro.

- **SR Pro – Printing**

Outlines the routines used to use one of the many different printing options provided by SuperReport Pro.

- **SR Pro – Developer Application Interface (API)**

Outlines the routines used to provide developer customization to the SuperReport Pro editor interface.

SR Pro – Access Routines

Each of the following routines provide the ability to load and save reports using the SuperReport Pro plug-in area, or outline the %SuperReport plug-in area

Routines included in this section are:

- *%SuperReport*
- SR Get Area
- SR Doc 2 Report
- SR Load Report
- SR Save Report
- SR Set Area
- SR New Report
- **SR New Offscreen Area**
- **SR DELETE OFFSCREEN**

%SuperReport

%SuperReport

Parameter	Data Type	Description
-----------	-----------	-------------

No Parameters

%SuperReport is the actual plug-in area object, which is placed on the 4th Dimension form in which you wish to display the SuperReport Pro area.

In addition, you use the %SuperReport routine when using the 4th Dimension **Open external window** routine, which provides the ability to display the SuperReport Pro editor in a standard 4D external window.

NOTE: For more information regarding the **Open external window** command, please refer to the 4th Dimension Language Reference Documentation.

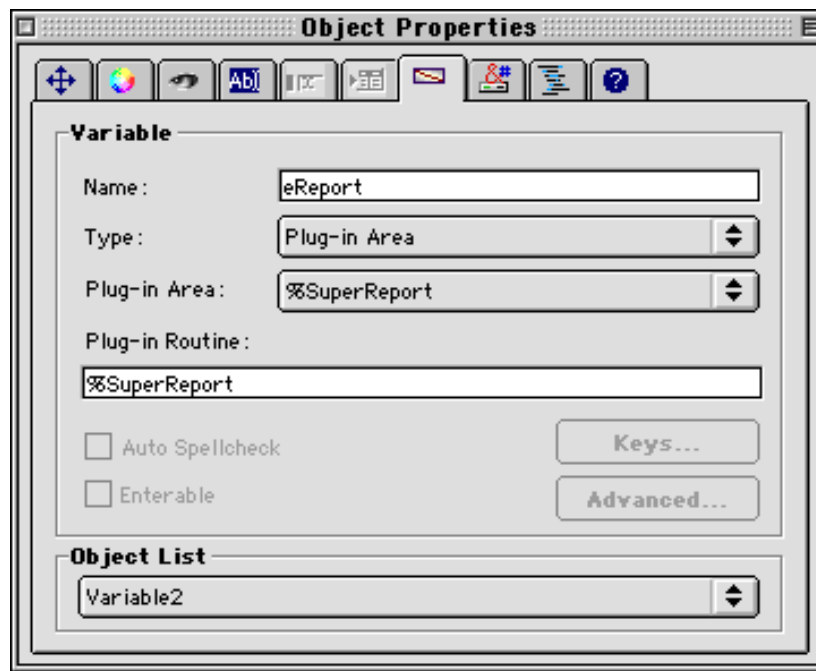


Figure 42 — 4D Object Properties Palette using %SuperReport area

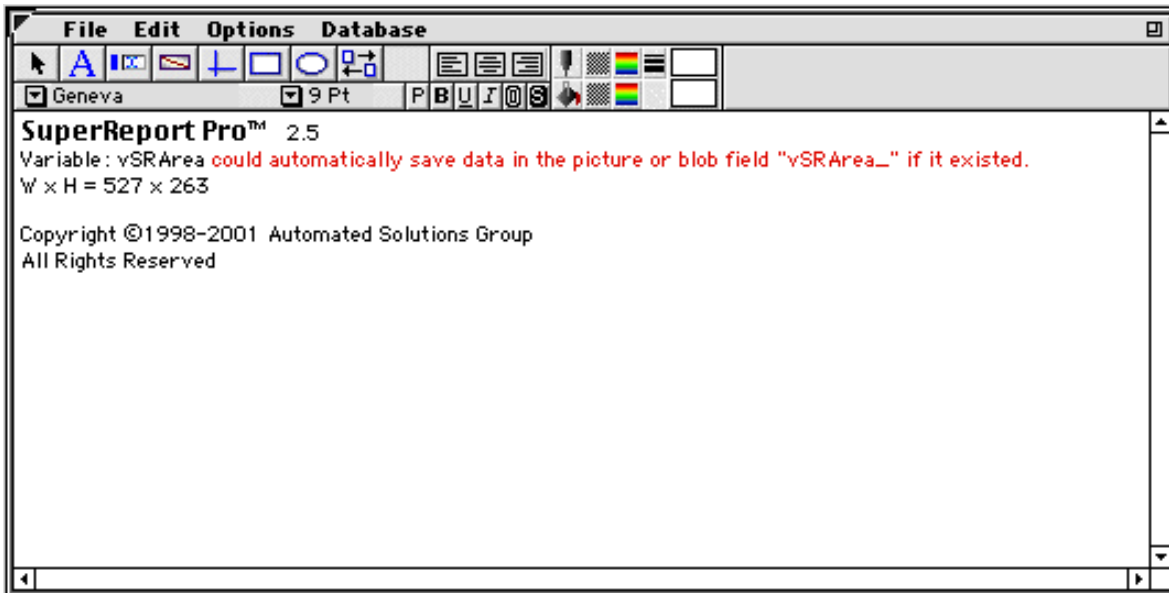


Figure 43 — SuperReport Pro Area on Form

Example:

The following example will display the SuperReport Pro report editor in a 4th Dimension external window using the Open external window routine.

C_LONGINT(iSRWindow) `don't use SRWindow

iSRWindow:=Open external window(50;50;Screen width-50;Screen height-50;8;"New Report";"%SuperReport")

SR Get Area

SR Get Area(reportArea:L; blobObject:P) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
BlobObject	C_BLOB	Result BLOB object
-> ResultCode	C_LONGINT	Result Code

SR Get Area will copy the information in the supplied reportArea and copy it to a 4th Dimension BLOB variable. You must use either a process or inter-process variable; using a 4th Dimension field will not return the information correctly.

ReportArea — The desired SuperReport Pro area.

BlobObject — A valid 4th Dimension variable which will receive the desired report information.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will extract the report in eReportArea and place it into a 4th DIMENSION process variable, then copy the information to a 4th Dimension BLOB field where the report is actually save.

C_BLOB(pReportData)

```
iSR_Err:=SR Get Area(eReportData;pReportData)
If (iSR_Err=0)
    [ReportMgr]ReportData:=pReportData
End if
```

See Also: SR Set Area

SR Set Area

SR Set Area(reportArea:L; reportData:P) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
ReportData	C_BLOB	Source BLOB object
-> ResultCode	C_LONGINT	Result Code

SR Set Area will take the reportData and place it in the defined SuperReport Pro area. This command will overwrite any existing data contained in the defined area. This routine is typically called in the forms **On Load** event before the form is displayed.

If you would like to determine if the current SuperReport area has been modified before overwriting the existing report, use the **SR Is Modified** routine.

ReportArea — The desired SuperReport Pro area.

ReportData — A valid 4th Dimension variable or field, which contains the reportData we wish to place in the defined SuperReport Pro area.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

WARNING: If you have customized the supporting file structure using the **SR Structure** routine, it **must** be called before calling **SR Set Area**.

Example:

The following example will place the report loaded from the Report Manager table and place it in eReportArea area, overwriting any existing report information.

Case of

```
:(Form event=On Load)
  iSR_Err:=SR Set Area(eReportArea;[ReportMgr]ReportData)
  If(iSR_Err#0)
    ALERT("An error occurred loading report data.")
  End if
```

End case

See Also: SR Get Area

SR Doc 2 Report

SR Doc 2 Report(reportDiskFile:S; resultBlobObject:P) -> resultCode:L

Parameter	Data Type	Description
ReportDiskFile	C_STRING(255)	Report Disk Filename
resultBlobObject	C_BLOB	Result BLOB object
-> ResultCode	C_LONGINT	Result Code

SR Doc 2 Report will load a document saved on disk and copy it to a 4th Dimension BLOB variable. You must use either a process or inter-process variable; using a 4th Dimension field will not return the information correctly. This command can be used to load a report without using the SuperReport Pro plug-in area.

ReportDiskFile — The full pathname to a valid SuperReport Pro report document, previously saved to disk using either the SuperReport Pro Editor, or via the **SR Save Report** routine.

BlobObject — A valid 4th Dimension variable which will receive the desired report information.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will load a SuperReport Pro document from disk and print it using the **SR Print Report** routine.

C_BLOB(pReportData)

C_TEXT(tReportDoc)

tReportDoc:="Hard disk:myReport.srp"

iSR_Err:=**SR Doc 2 Report**(tReportDoc;pReportData)

If(iSR_Err=0)

 iSR_Err:=**SR Print Report**(pReportData;0;0) `print the report

End if

SR Load Report

SR Load Report(reportArea:L; reportDiskFile:S) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
ReportDiskFile	C_STRING(255)	Report Disk File
-> ResultCode	C_LONGINT	Result Code

SR Load Report will load a document saved on disk and place it in the supplied SuperReport Pro area. This command is equivalent to selecting **Open from Disk** menu item from the **File** menu in the SuperReport Pro report area.

ReportArea — The desired SuperReport Pro area.

ReportDiskFile — The full pathname to a valid SuperReport Pro report document, previously saved to disk using either the SuperReport Pro Editor, or via the **SR Save Report** routine. If the *ReportDiskFile* parameter is a null string, the standard Open File dialog will be display, allowing the user to locate the file manually.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will load a SuperReport Pro document from disk and place in the **eReportArea**.

```
C_TEXT(tReportDoc)
```

```
tReportDoc:="Hard disk:myReport.srp"
```

```
iSR_Err:=SR Load Report(eReportArea;tReportDoc) `use the tReportDoc report
```

```
iSR_Err:=SR Load Report(eReportArea;") `display the Open File dialog
```

See Also: SR Load Report

SR Save Report

SR Save Report(reportArea:L; reportDiskFile:S; options:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
ReportDiskFile	C_STRING(255)	Report Disk File
Options	C_LONGINT	Report Options
-> ResultCode	C_LONGINT	Result Code

SR Save Report will save the contents of the defined SuperReport Pro report area to a standard disk file. This command is equivalent to selecting **Save to Disk** menu item from the **File** menu in the SuperReport Pro report area.

ReportArea — The desired SuperReport Pro area.

ReportDiskFile — The full pathname to a valid SuperReport Pro report document.. If the *ReportDiskFile* parameter is a null string, the standard Save File dialog will be display, allowing the user to define the location of the file manually.

Options — Controls the display of the Save File dialog.

0 — No custom action performed

1 — Display the Save File dialog, regardless of the supplied pathname.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will save a SuperReport Pro document to disk, which is contained in the **eReportArea**.

C_TEXT(tReportDoc)

tReportDoc:="Hard disk:myReport.srp"

iSR_Err:=**SR Save Report**(eReportArea;0;tReportDoc) `use the tReportDoc report

iSR_Err:=**SR Load Report**(eReportArea;1;tReportDoc) `display the Open File dialog

See Also: SR Save Report

SR New Report

SR New Report(resultBlobObject:P) -> resultCode:L

Parameter	Data Type	Description
ResultBlobObject	C_BLOB	Result BLOB object
-> ResultCode	C_LONGINT	Result Code

SR New Report will create a new report and place it in the supplied report variable. This routine should be called whenever you are creating a new report, or display a dialog, which contains a SuperReport Pro area and you have not called the **SR Set Area** routine to preload the area with an existing report.

ReportData — A valid 4th Dimension variable, which will contain the new report.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

NOTE: If you don't call **SR New Report**, SuperReport Pro attempts to create a new report when the form is displayed. If the report area does not display the report area grid, etc. you must call **SR New Report** to properly create the new area.

Example:

The following example creates a new report, then place it in the supplied SuperReport Pro area, using the **SR Set Area** routine.

C_BLOB(pReportData)

Case of

```
:(Form event=On Load)
  iSR_Err:=SR New Report(pReportData)
  iSR_Err:=SR Set Area(eReportArea;pReportData)
  If(iSR_Err#0)
    ALERT("An error occurred creating the report.")
  End if
```

End case

SR New Offscreen Area

SR New Offscreen Area -> reportObject:L

Parameter	Data Type	Description
None		
-> ReportObject	C_LONGINT	SuperReport Pro Object Reference

SR New Offscreen Area will create a new report object in memory, which can be used to in conjunction with other SuperReport Pro routines for procedurally creating reports. When you have finished using the offscreen area, you must destroy the object using **SR DELETE OFFSCREEN AREA**.

-> *Report Object*—If the routine completed successfully, a valid SuperReport Pro object will be created. If the report was not created, a valid SuperReport Pro error code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example is the basic construct for procedurally creating reports. For more information, please refer to **Chapter 5 — Tutorial (Procedurally Creating Reports)** or **Chapter 6 — Command Reference (Developer Interface [API])**.

C_LONGINT(\$SR_AREA)

```
$SR_AREA:=SR New Offscreen Area
If ($SR_AREA>0)
  `create objects
  `save or print report
  SR DELETE OFFSCREEN AREA($SR_AREA)
End if
```

See Also: SR DELETE OFFSCREEN AREA

SR DELETE OFFSCREEN AREA

SR DELETE OFFSCREEN AREA(ReportObjectRef:L)

Parameter	Data Type	Description
ReportObjectRef	C_LONGINT	Valid offscreen area

SR DELETE OFFSCREEN AREA will destroy the report object reference, created using **SR New Offscreen Area**.

Report Object Reference —A valid offscreen report reference, created using **SR New Offscreen Area**.

Example:

See **SR New Offscreen Area** for example of using this routine

See Also: SR New Offscreen Area

SR Pro – Customizing

Each of the following routines provides the ability to customize the SuperReport Pro user access variables or file structure, as well as report editor and dialog customization routines.

Routines included in this section are:

- **SR Get Area Format**
- **SR Set Area Format**
- *SR Options*
- **SR Get Options**
- **SR Set Options**
- SR Std Vars
- *SR Structure*
- SR Variables
- *SR Commands*

SR Get Area Format

SR Get Area Format(reportArea:L; areaFormat:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
AreaFormat	C_LONGINT	Report area format
-> ResultCode	C_LONGINT	Result Code

SR Get Area Format will return the report pages format, either physical page or printable area.

ReportArea — The desired SuperReport Pro area.

AreaFormat — Returns the report area format. One of the following values will be returned.

1 - [SR AreaFormat Physical Page](#)

2 - [SR AreaFormat Printable Area](#)

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

See Also: SR Set Area Format

Example:

The following example will return the report area format for a SuperReport Pro area

```
$SR_Err:=SR Get Area Format($SR_AREA;$SR_AreaFormat)
```

See Also: SR Set Area Format

SR Set Area Format

SR Set Area Format(reportArea:L; areaFormat:L; adjustOptions:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
AreaFormat	C_LONGINT	Report area format
AdjustOptions	C_LONGINT	Report adjustment options
-> ResultCode	C_LONGINT	Result Code

SR Set Area Format will set the report pages format, either physical page or printable area and perform any specified formatting options. This routine performs the same task as selecting **Options | Physical Page** or **Options | Printable Area** when using the SuperReport Pro editor.

ReportArea — The desired SuperReport Pro area.

AreaFormat — The desired area format you wish to set for the defined report area. You may use one of the following values.

- 1 - [SR AreaFormat Physical Page](#)
- 2 - [SR AreaFormat Printable Area](#)

AdjustOptions — When changing the area format, you may choose to perform automatic adjustments of the report objects. You may choose one or more of the following values.

- 1 - [SR SetAreaFormat Adjust Objects](#)
- 2 - [SR SetAreaFormat Adjust Sects](#)
- 4 - [SR SetAreaFormat Adjust Guides](#)

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will return the report area format for a SuperReport Pro area

```
$adjustOpts=\_SR SetAreaFormat Adjust Objects | SR SetAreaFormat Adjust Sects | SR SetAreaFormat Adjust Guides  
$SR_Err:=SR Set Area Format($SR_AREA; SR AreaFormat Physical Page; $adjustOpts)
```

See Also: SR Get Area Format

SR Options

SR Options(reportArea:L; showReportMenu:L; enableScriptAccess:L; miscReportOpts:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
ShowReportMenu	C_LONGINT	Show "Database" menu
EnableScriptAccess	C_LONGINT	Enable access to Scripts button
Miscellaneous Opts	C_LONGINT	Additional configuration options
-> ResultCode	C_LONGINT	Result Code

SR Options allows you to customize the display and configuration of a SuperReport Pro area.

This routine is identical to the **SR Set Options** routine and remains for backwards compatibility with older versions of SuperReport Pro. If you are using routine for the first time, we recommend you use the **SR Set Options** routine.

ReportArea — The desired SuperReport Pro area.

ShowReportMenu — Configures the display of the "Database" menu (last menu in the report editor's menu bar). The default value is 1.

- 0 (SR Generic Option Set Off) – Don't display "Database" menu bar
- 1 (SR Generic Option Set On) – Display "Database" menu bar (default value)

EnableScriptAccess — Configures access the various Script buttons and menu items. Script access controls all places where script access is available.

- 0 (SR Generic Option Set Off) – Disable access to report scripts.
- 1 (SR Generic Option Set On) – enable access to report scripts

Miscellaneous Options — Configures additional miscellaneous options. This parameter is a standard bitwise value, you simply add the options together and pass it as single to value to activate one or more options.

- 0 (SR Generic Option Set Off) – No miscellaneous options (default value)
- 1 (SR Options Hide Zoom) - Hide Zoom box and disable Zoom Window menu item
- 2 (SR Options Basic Interface) - Use basic Select Main table dialog
- 4 (SR Options Hide HTML) - Disable HTML access (HTML button will be removed from appropriate dialogs)
- 8 (SR Options Use Std Guides) - Use standard guides when draw report area (red, blue, etc.)
- 32 (SR Options Hide Menubar) – Hide Menubar
- 64 (SR Options Hide Toolbar) – Hide Toolbar
- 128 (SR Options Hide Hor Scrollbar) – Hide Horizontal Scrollbar
- 256 (SR Options Hide Vert Scrollbar) – Hide Vertical Scrollbar

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

NOTE: Passing a value of -1 on any of the option parameters will use the current settings.

See Also: SR Set Options, SR Get Options

SR Get Options

SR Get Options(reportArea:L; showReportMenu:L; enableScriptAccess:L; miscReportOpts:L; zoomTitle) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
ShowReportMenu	C_LONGINT	Show “Database” menu
EnableScriptAccess	C_LONGINT	Enable access to Scripts button
Miscellaneous Opts	C_LONGINT	Additional configuration options
Zoom Title	C_STRING	Zoomed window title
-> ResultCode	C_LONGINT	Result Code

SR Get Options allows you to retrieve the custom options (see **SR Set Options**) a SuperReport Pro area.

ReportArea — The desired SuperReport Pro area.

ShowReportMenu — A valid 4th Dimension variable, which will receive the setting of the “Database” menu.

0 (SR Generic Option Set Off) – Don’t display “Database” menu bar

1 (SR Generic Option Set On) – Display “Database” menu bar (default value)

EnableScriptAccess — A valid 4th Dimension variable, which will receive the “Script” access status. Script access controls all places where script access is available.

0 (SR Generic Option Set Off) – Disable access to report scripts.

1 (SR Generic Option Set On) – enable access to report scripts

Miscellaneous Options — A valid 4th Dimension variable, which will receive the bitwise value of the miscellaneous options.

0 (SR Generic Option Set Off)– No miscellaneous options (default value)

1 (SR Options Hide Zoom) - Hide Zoom box and disable Zoom Window menu item

2 (SR Options Basic Interface) - Use basic Select Main table dialog

4 (SR Options Hide HTML) - Disable HTML access (HTML button will be removed from appropriate dialogs)

8 (SR Options Use Std Guides) - Use standard guides when draw report area (red, blue, etc.)

32 (SR Options Hide Menubar) – Hide Menubar

64 (SR Options Hide Toolbar) – Hide Toolbar

128 (SR Options Hide Hor Scrollbar) – Hide Horizontal Scrollbar

256 (SR Options Hide Vert Scrollbar) – Hide Vertical Scrollbar

Zoomed Title — A valid 4th Dimension variable, which will receive the customized window title which may have been set using **SR Set Options**.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

See Also: SR Set Options

SR Set Options

SR Set Options(reportArea:L; showReportMenu:L; enableScriptAccess:L; miscReportOpts:L; zoomedWindowTitle:S) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
ShowReportMenu	C_LONGINT	Show “Database” menu
EnableScriptAccess	C_LONGINT	Enable access to Scripts button
Miscellaneous Opts	C_LONGINT	Additional configuration options
Zoomed Win Title	C_STRING	Zoomed window title
-> ResultCode	C_LONGINT	Result Code

SR Set Options allows you to customize the display and configuration of a SuperReport Pro area.

ReportArea — The desired SuperReport Pro area.

ShowReportMenu — Configures the display of the “Database” menu (last menu in the report editor’s menu bar). The default value is 1.

- 0 ([SR Generic Option Set Off](#)) – Don’t display “Database” menu bar
- 1 ([SR Generic Option Set On](#)) – Display “Database” menu bar (default value)

EnableScriptAccess — Configures access the various Script buttons and menu items. Script access controls all places where script access is available.

- 0 ([SR Generic Option Set Off](#)) – Disable access to report scripts.
- 1 ([SR Generic Option Set On](#)) – enable access to report scripts

Miscellaneous Options — Configures additional miscellaneous options. This parameter is a standard bitwise value, you simply add the options together and pass it as single to value to activate one or more options.

- 0 ([SR Generic Option Set Off](#))– No miscellaneous options (default value)
- 1 ([SR Options Hide Zoom](#)) - Hide Zoom box and disable Zoom Window menu item
- 2 ([SR Options Basic Interface](#)) - Use basic Select Main table dialog
- 4 ([SR Options Hide HTML](#)) - Disable HTML access (HTML button will be removed from appropriate dialogs)
- 8 ([SR Options Use Std Guides](#)) - Use standard guides when draw report area (red, blue, etc.)
- 32 ([SR Options Hide Menubar](#)) – Hide Menubar
- 64 ([SR Options Hide Toolbar](#)) – Hide Toolbar
- 128 ([SR Options Hide Hor Scrollbar](#)) – Hide Horizontal Scrollbar
- 256 ([SR Options Hide Vert Scrollbar](#)) – Hide Vertical Scrollbar

Zoom Window Title — Configures the window title of a zoomed area from editor.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

NOTE: Passing a value of –1 on any of the option parameters will use the current settings.

See Also: SR Get Options

Example:

The following example will disable access to scripts.

C_BLOB(pReportData)

Case of

```
:(Form event=On_Load)
  iSR_Err:=SR New Report(pReportData)
  iSR_Err:=SR Set Area(eReportArea;pReportData)
  iSR_Err:=SR Set Options(eReportArea;-1;-1)
```

End case

The following example will disable the zoom window, and disable access to HTML buttons.

C_BLOB(pReportData)

Case of

```
:(Form event=On_Load)
  iSR_Err:=SR New Report(pReportData)
  iSR_Err:=SR Set Area(eReportArea;pReportData)
  iSR_Err:=SR Set Options(eReportArea;-1;-1;SR Options Hide Zoom + SR Options Hide HTML)
```

End case

SR Std Vars

SR Std Vars(whichVar:L; varName:S) -> resultCode:L

Parameter	Data Type	Description
Which Var	C_LONGINT	Which SR Pro variable
Variable Name	C_STRING(32)	Replacement variable name
-> ResultCode	C_LONGINT	Result Code

SR Std Vars allows you to replace the standard SuperReport Pro “reporting” variables (SRDate, SRTime, etc.) with ones of your own choosing. When defining a replacement variable, make sure the data types are the same as the original SR Pro variable and that your replacement variable has been properly declared and initialized.

Which Variable — Which of the 4 SR Pro variables you wish to override.

- 1 – SRDate (C_DATE)
- 2 – SRTime (C_LONGINT)
- 3 – SRPage (C_LONGINT)
- 4 – SRRecord (C_LONGINT)

Variable Name — A valid 4th Dimension variable which will be used as a replacement for the defined SR Pro variable.

NOTE: This command replaces the standard variables for all SuperReport Pro areas and reports printed by the 4D database application using SuperReport Pro.

Example:

The following example will override the default date variable (SRDate) with our own date variable.

C_DATE(dMyDateVar)

Case of

```
:(Form event=On_Load)
  iSR_Err:=SR Std Vars(1;"dMyDateVar")
```

End case

SR Structure

SR Structure(reportArea:L; structureArrayName:S; options:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Structure Array Name	C_STRING(255)	Name of array containing structure
Options	C_LONGINT	Options
-> ResultCode	C_LONGINT	Result Code

SR Structure allows you to override the default file structure with your own custom database structure view. By default, SuperReport Pro will display the first 128 visible tables, and will list all the visible fields for each table displayed.

If you have customized your structure using 4Ds SET TABLE TITLES and SET FIELD TITLES, this information will be used by default and calling SR Structure is an additional interface for creating the custom structure.

Calling SR Structure with the appropriate options (see *Options* below) will override the virtual structure you may have created using the 4D virtual structure commands.

When configuring a custom structure view, you can not only limit access to only those fields which you wish the user to have access; you can also provide a field “alias” which will be displayed when the user selects the Field selection popup menu.

WARNING: If you have customized the supporting file structure using the **SR Structure** routine, it **must** be called before calling **SR Set Area**.

ReportArea — The desired SuperReport Pro area.

Structure Array Name — The name (this is a string parameter, not the actual array) of the array, which contains the new structure view, using the SuperReport Pro structure format:

<menu number> ; <table number>; <field number>; <virtual name>

menu number – allows you to build a hierarchical menu

table number – the desired 4th Dimension table number you wish to use

field number – the corresponding 4th Dimension field number you wish to use. Passing a value of 0 will include all fields for the defined table.

virtual name – the “alias” name for the desired table or fieldname

Failure to supply the table name (parent menu name) will result in the creation of a menu that contains a blank parent menu and SuperReport Pro will not be able to correctly create the table list used by Main Table/Iterations in the SuperReport Pro Editor, therefore you need to make sure the table name is supplied when creating the parent menu entry.

Options — Structure definition options. You may use one or more of the following values (***bold italic*** items are on by default):

0 (SR Structure Physical) — Original SuperReport Pro 2.0.x method

1 (*SR Structure Virtual*) – Use 4Ds Virtual Structure settings, otherwise structure values will be used

4 (SR Structure Get Invisible) – Allow use of invisible tables and fields

16 (SR Structure Get Indexed Only) — Only use indexed tables/fields

32 (SR Structure Get Empty Tables) — Include tables which contain zero fields in list (otherwise omitted)

64 (*SR Structure Sort By Name*) — Sort the tables and fields in alphabetical order

128 (SR Structure No MenuID) — Used when creating a custom structure for Table list, SRP ignores menuID

256 (*SR Structure Mark Indexed*) — When structure is displayed, indexed columns will be bold

512 (SR Structure Mark Invisible) — When structure is displayed, invisible fields will be *italic*

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

This following example will produce a hierarchical menu with 2 sub menus; one for the Supplier table, and the second with two fields from the Customer table.

C_BLOB(pReportData)

Case of

:(Form event=On Load)

ARRAY STRING(64;aSR_Structure;5)

aSR_Structure{1}:="1;1;0;Suppliers" `name menu 1 "Suppliers" this could be an alias

aSR_Structure{2}:="1;1;0" `add all fields for table 1

aSR_Structure{3}:="2;5;0;Customers" `name menu 2 "Customers"

aSR_Structure{4}:="2;5;1;Customer Code"

aSR_Structure{5}:="2;5;3;Customer Name"

iSR_Err:=**SR Structure**(eReportArea;"aSR_Structure")

iSR_Err:=**SR New Report**(pReportData)

iSR_Err:=**SR Set Area**(eReportArea;pReportData)

iSR_Err:=**SR Options**(eReportArea;-1;1;-1)

End case

See Also: SR Commands, SR Variables

SR Variables

SR Variables(reportArea:L; variableArrayName:S) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
VariableArrayName	C_STRING(255)	Name of array containing variables
-> ResultCode	C_LONGINT	Result Code

SR Variables allows you customize the list of available variables in the Variable dialog for the defined SuperReport Pro area. Using this command, you can provide users access to any variable within your application.

When using custom variables, you must properly declare the variables before they are used in the SuperReport Pro editor.

If you wish to clear the variables pop-up menu, then either pass an empty string as the array name or size the array to zero elements and execute the command again. When removing custom variables, the variable list will only contain the default SuperReport Pro variables.

Note: There is a limit of limit of 31 parent menus when creating custom variable submenus.

ReportArea — The desired SuperReport Pro area.

Variable Array Name — The name (this is a string parameter, not the actual array) of the array which contains the list of variables, using the following format:

<menu number> ; <menu/item name>; <variable name>; <variable type>; <array element>

menu number – the desired menu item which you wish to add the variable

menu/item number – the name of the menu or item.

variable name – the name of the corresponding 4th Dimension variable.

variable type – the type of variable.

1 – standard 4th Dimension variable

2 – automatic array (uses SRRecord as index item)

3 – specified array element

array element – if you have specified a variable type of 3, this item will contain the specific array element

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following example will produce a hierarchical menu with 2 sub menus; one for the standard SuperReport Pro variables, and the second for application specific variables.

C_BLOB(pReportData)

Case of

:(Form event=On Load)

ARRAY STRING(64;aSR_Variables;5)

aSR_Variables {1} := "1;Standard Variables" `name menu 1 "Standard Variables"

aSR_Variables {2} := "1;Date;SRDate;1" `add a variable to menu 1

aSR_Variables {3} := "1;Time;SRTime;1" `add a variable to menu 1

aSR_Variables {4} := "1;Page Number;SRPage;1" `add a variable to menu 1

aSR_Variables {5} := "1;Record;SRRecord;1" `add a variable to menu 1

aSR_Variables {6} := "2;System Variables" `name menu 2 "Standard Variables"

aSR_Variables {7} := "2;Username;<>vUser;1" `add a variable to menu 2

aSR_Variables {8} := "2;Month Names; <>aMonths;2" `add an array item (automatic)

aSR_Variables {7} := "2;Monday;aWkDays;3;2" `add a specified array element

\$result:= **SR Variables** (eReportData; "aSR_Variables")

iSR_Err:=**SR New Report**(pReportData)

iSR_Err:=**SR Set Area**(eReportArea;pReportData)

End case

See Also: SR Commands, SR Structure

SR Commands

SR Commands(reportArea:L; commandArrayName:S; options:L; excludeThemesArrayName:S; excludeCommandsName:S) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
commandArrayName	C_STRING(255)	Name of array containing commands
-> ResultCode	C_LONGINT	Result Code

SR Commands allows you customize the list of available commands in the Script dialog for the defined SuperReport Pro area. Using this command, you can provide users access to specific commands and application routines, without exposing the entire 4th Dimension Command list.

If you don't supply a command array name, you still have control over the way in which SuperReport Pro builds the command list. When using an empty string as the *arrayName* parameter, you may choose to use one or more of the additional formatting options.

NOTE: You can use standard Macintosh meta-characters in the command array to better distinguish the routines. See the example below for more information.

ReportArea — The desired SuperReport Pro area.

Command Array Name — The name (this is a string parameter, not the actual array) of the string array, which contains the list of commands. Passing a null string will cause SuperReport Pro to build the command list use the current list of 4D Commands.

Options — When passing an empty string as the array name, SuperReport Pro will build the command list based on one of the following conditions using the entire 4D Command List. You may choose one or more of the following values.

0 (SR Commands By First Char) – Group commands by first character of routine name

1 (SR Commands By Theme) – Group commands by theme name

4 (SR Commands Sort Second Level) – Sort commands by second level

8 (SR Commands Sort First Level) – Sort first level (when grouping by first char, first level is always sorted)

When building the command list, you can conditionally format the list items by using one of the following meta-characters.

<B – applies the bold attribute to the list item

<I – applies the italic attribute to the list item

<U – applies the underline attribute to the list item

Note: With the release of SuperReport Pro v2.8, command lists (and other related lists) are displayed using 4D style hierarchical lists, thus the use of other meta-characters are not supported and will ignored.

ExcludeThemeArray— The name (string parameter) of a 4th Dimension integer or longint array containing the STR# 4 index theme items you wish to exclude from the command list.

ExcludeCommandArray — The name (string parameter) of a 4th Dimension integer or longint array containing the STR# 8 index command items you wish to exclude from the command list.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following example will create a custom array of commands names, using meta-characters for custom functions and plug-in routines.

Case of

```
:(Form event=On Load)
  ARRAY STRING(63;aSR_Commands;5)
  aSR_Commands {1}:="ALL RECORDS"
  aSR_Commands {2}:="FIRST RECORD"
  aSR_Commands {3}:="myFunction<I"
  aSR_Commands {4}:="MT ALERT<B<I"
  aSR_Commands {5}:="Z Command"
  $result:=SR Commands (eReportArea;"aSR_Commands")
```

End case

The following will create the command list based on the 4D commands, grouped by theme names and will be an unsorted list.

Case of

```
:(Form event=On Load)
  $result:=SR Commands (eReportArea;""; SR Commands By Theme)
```

End case

The following example will create the command list based on 4D commands, sorting the list by Theme name and excluding compiler directives.

Case of

```
:(Form event=On Load)
  ARRAY LONGINT(aiSR_Theme;1)
  aiSR_Theme{1}:="7 ` Compiler"
  $opts:= SR Commands By Theme | SR Sort First Level | SR Sort Second Level
  $result:=SR Commands (eReportArea;"";$opts;"aiSR_Theme")
```

End case

The following example will build on the previous example, and include a list of individual routines, which should be omitted (any delete record action) from the command list. The result list will be sorted by Theme name, and each routine will be in alphabetical order within it respective theme.

Case of

```
:(Form event=On_Load)
  ARRAY LONGINT(aiSR_Theme;1)
  aiSR_Theme{1}:=7 `Compiler
  ARRAY LONGINT(aiSR_Command;3)
  aiSR_Command{1}:=58
  aiSR_Command{2}:=66
  aiSR_Command{3}:=96
  $result:=SR Commands (eReportArea;""; SR Commands By Theme + SR Commands By First Char;
    "aiSR_Theme";"aiSR_Command")
```

End case

See Also: SR Variables, SR Structure

SR Pro – Menus

Each of the following routines provide the ability to customize the text and associated actions for each of the SuperReport Pro editor menus, as well as the ability to procedurally execute any of the available menu items as if the actual menu item was selected.

Routines included in this section are:

- SR Do Command
- SR Menu Info
- SR Menu Item

SR Do Command

SR Do Command(reportArea:L; menuItem:L; option:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
SRProMenu	C_LONGINT	Desired Menu Item
Option	C_LONGINT	Menu execution option
-> ResultCode	C_LONGINT	Result Code

SR Do Command allows you procedurally execute any of the available menu items as if the user selected the menu item for the desired SuperReport Pro menu. If the menu item is disabled, this command will have no affect.

If a custom 4th Dimension method has been defined using SR Menu Item, this method will be executed unless you pass a value of 1 using the *Option* parameter.

ReportArea — The desired SuperReport Pro area.

SRProMenu — The internal SuperReport Pro menu number, which you wish to work with. For a complete list of menu codes, please refer to **Chapter 7 — SuperReport Pro Codes (Menu Numbers)**.

Option — Menu execution option

0 – default action, executes 4th Dimension method if it has been supplied using **SR Menu Item**, otherwise default action will be executed.

1 – Execute SuperReport Pro default action, regardless of override method assigned using **SR Menu Item**.

0 - SR DoCommand Use Custom Proc

1 - SR DoCommand Ignore Custom Proc

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following example will execute the Print Preview command, from the File menu. Notice the option parameter has a value of 1, this will ensure that the standard SuperReport Pro action will be executed, despite any override method assigned via SR Menu Item.

```
iSR_Err:=SR Do Command(eReportArea;107;1) `execute the default Print Preview action
```

See Also: SR Menu Info, SR MenuItem

SR Menu Info

SR Menu Info(reportArea:L; SRProMenu:L; menuID:L; menuItem:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
SRProMenu	C_LONGINT	SuperReport Pro Menu
MenuID	C_LONGINT	Desired Menu
ItemID	C_LONGINT	Desired Menu Item
-> ResultCode	C_LONGINT	Result Code

SR Menu Info will return the internal SuperReport Pro SRProMenu for a given menuID and itemID, or the menuID and itemID for a given SRProMenu.

- If you wish to return the SRProMenu, pass assign a value of -1 as the SRProMenu before calling and pass valid menuID and itemID values.
- If you wish to return the MenuID and itemID, assign values of -1 to each variable before calling and pass a valid SRProMenu.

ReportArea — The desired SuperReport Pro area.

SRProMenu — The internal SuperReport Pro menu number, which you wish to work with. See SuperReport Codes for a complete list of available SuperReport Pro Menu Numbers.

MenuID — The desired SuperReport Pro menuID. The menu numbers are numbered from 0 (Apple menu) to 4 (Database menu).

ItemID — The desired SuperReport Pro menu item. The menu item numbers begin with 1 (first item) and you should account for any separator or disabled items when using this parameter.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following example will return the *SRProMenu* for the desired *menuID* and *itemID*, in this we are returning the *menuID* and *menuItem* for the Print Preview menu item.

C_LONGINT(iSRProMenu;iMenuID;iItemID)

```
iSRProMenu:=-1  
iMenuID:=2      `edit menu  
iItemID:=3     `cut command
```

```
iSR_Err:=SR Menu Info(eReportArea;iSRProMenu;iMenuID;iItemID)      `returns iSRProMenu
```

The following example will return the menuID and itemID for SRProMenu

```
iSRProMenu:=107  
iMenuID:=-1  
iItemID:=-1
```

```
iSR_Err:=SR Menu Info(eReportArea;iSRProMenu;iMenuID;iItemID)      `returns iMenuID and iItemID
```

See Also: SR Do Command, SR MenuItem

SR Menu Item

SR Menu Item (reportArea:L; action:L; menuID:L; text:S; enable:L; mark:L; overrideMethod:S) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Action	C_LONGINT	Menu item action
SRProMenu	C_LONGINT	SuperReport Pro MenuID
Text	C_STRING(32)	Menu item text
Enable	C_LONGINT	Menu item enable flag
Mark	C_STRING(2)	Menu item mark
Override Method	C_STRING(32)	Replacement Method
-> ResultCode	C_LONGINT	Result Code

SR Menu Item is used to query a menu item's characteristics and attributes, or to set one or more of the menu items attributes (such as the menu item text). This command can also be used to override the default menu action with a custom 4th Dimension method.

ReportArea — The desired SuperReport Pro area.

Action — Desired action to perform, using the following action codes:

You may choose one of the following

-1 - [SR MenuItem Count Items](#)

0 - [SR MenuItem Query](#)

Or combination of the following:

1 - [SR MenuItem Set Text](#)

2 - [SR MenuItem Set Status](#)

4 - [SR MenuItem Set Mark](#)

8 - [SR MenuItem Set 4D Method](#)

0 – Query desired menuID

1 – Set menu item text (using Text parameter)

2 – Set menu item status (using Enable parameter)

4 – Set menu item mark (using Mark parameter)

8 – Set override method (using Override Method parameter)

SRProMenu — The internal SuperReport Pro menu number, which you wish to work with. For a complete list of SuperReport Pro MenuIDs, please refer to **Chapter 7 – SuperReport Pro Codes (Menu Numbers)**

Text — The desired menu item text for the defined SuperReport Pro menu number.

Enable Flag — Enable flag for the defined SuperReport Pro menu number.

0 – Disable menu item

1 – Enable menu item

Mark — The desired menu item mark for the defined SuperReport Pro menu number.

0 – No mark

18 – Check mark

19 – Diamond mark

NOTE: When using this routine on Windows to set the Mark parameter, only the checkmark value will be used, all other characters area ignored.

Override Method — The desired 4th Dimension method which will be executed when the menu item is selected. If you wish to clear the override method, pass a null string (“”) as the Override Method parameter.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following example will replace the default action of the New command on the File menu to execute the *MyDoNew* method.

C_LONGINT(iSRProMenu;iEnabled)

C_STRING(255;sItemText;sMark;sProcName)

iSRProMenu:=101 `FileNew

sItemText:=”

sMark:=”

sProcName:=”

iSR_Err:=**SR Menu Item**(eReportArea;0;iSRProMenu;sItemText;iEnabled;sMark;sProcName)

If (iSR_Err=0) `make sure no error occurred

If (sProcName=”) `see if the default method is current assigned

 iSR_Err:=**SR Menu Item**(eReportArea;2+8;iSRProMenu;””;1;0;”MyDoNew”)

End if

End if

End if

`GP: MyDoNew

`DC: Example override method for SuperReport Pro

iSR_Err:=**SR New Report**(eReportArea)

See Also: SR Do Command, SR Menu Info

SR Pro – Miscellaneous

Each of the following routines provides miscellaneous actions when interacting with the SuperReport Pro report editor.

Routines included in this section are:

- SR ABOUT
- SR Register
- SR Document
- SR File Types
- **SR Get Format Number**
- **SR Get Table List**
- **SR Get Field List**
- **SR Get Commands As List BLOB**
- **SR Get Tables As List BLOB**
- **SR Get Fields As List BLOB**
- **SR Get Structure As List BLOB**
- **SR Get Variables As List BLOB**
- **SR Get Variable Type**
- **SR Get Virtual Variable**
- **SR Get Indexed String**
- SR Info
- SR Is Modified
- SR Set Modify
- *SR Main Table*
- *SR Main Table2*
- SR On Event
- **SR Power Menu**
- SR RELATIONS
- **SR Get Tips Enabled**
- **SR SET TIPS ENABLED**
- **SR SWAP HANDLES** (deprecated)
- **SR Report To BLOB**

SR ABOUT

SR ABOUT

SR ABOUT Displays the SuperReport Pro about dialog.



Figure 44 — SuperReport Pro About Box

SR Register(registrationKey:S) -> resultCode:L

Parameter	Data Type	Description
registrationKey	C_STRING(80)	Registration Code
-> resultCode	C_LONGINT	Result Code

SR Register is used to register the SuperReport Pro plug-in for standalone or server use. You must call **SR Register** with a valid registration key; otherwise SuperReport Pro will operate in demonstration mode. Without a valid registration key, SuperReport Pro will operate in demonstration mode during 20 minutes.

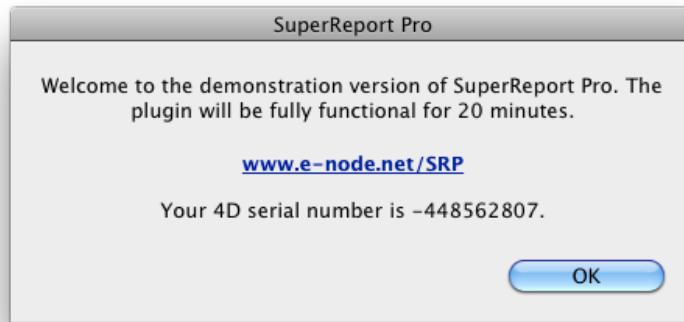


Figure 45 — SuperReport Pro Demonstration mode Box

Like all e-Node plug-ins, SuperReport Pro offers six different license types. There are no such things as MacOS vs Windows or Development vs Deployment:

Single user license. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on 4D Standalone or Runtime. Since the registration key is linked to a specific 4D license, you need to provide the number returned by the 4D command **GET SERIAL INFORMATION** (first parameter). A new license will be supplied for free at any time if you change your 4D version and / or get a new 4D registration key, provided that your previous licenses match the current public version at the exchange time.

Small server. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on 4D Server up to 10 users. The registration key is linked to your 4D Server license just as above.

Medium server. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on 4D Server up with 11 to 20 users. The registration key is linked to your 4D Server license just as above.

Large server. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on 4D Server over 20 users. The registration key is linked to your 4D Server license just as above.

Unlimited Single User. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on as many 4D Standalone, Runtime or Engine copies that run your 4D application(s). This is a yearly license, which expires after the date when it is to be renewed. The expiration only affects interpreted mode. **Compiled applications using an obsolete license will never expire.**

Unlimited OEM. This license allows development (interpreted mode) or deployment (interpreted or compiled mode) on as many 4D Server (of any umber of users), 4D Standalone, Runtime or Engine copies that run your 4D application(s). This is a yearly license, which expires after the date when it is to be renewed. The expiration only affects interpreted mode. **Compiled applications using an obsolete license will never expire.**

A 4D database used to retrieve your 4D serial information is available from the following link:

<http://www.e-node.net/ftp/GetSerialInfo>

NOTE: The registration system has been modified in version 2.9. Only one registration key is now required.

registrationKey — Pass the registration key to register your copy of SuperReport Pro. Only one registration key is required. The key is either linked to the 4D or 4D Server serial number (individual licenses), or to the name of the company / developer (unlimited annual licenses).

-> *resultCode* — This will return a value of 1 if the registration key is valid and a value of 0 if the registration key is invalid. You should verify the correctness of the registration key by tracing over the call to **SR Register** and examining *resultCode*.

NOTE: Multiple calls to **SP Register** are allowed. The plug-in will be activated if at least one valid key is used.

The first successful call will return 0 and all subsequent **SR Register** calls will be ignored (and return 0).

Example:

```
C_LONGINT ($result)
$result:=SR Register ("Place your registration key here")
If ($result#1) `error
    ALERT ("SuperReport Pro could not be registered: "+String ($result))
End if
```

Example with multiple calls:

```
C_LONGINT ($result) `ignored in this case
$result:= SR Register ("Registration key one")
$result:= SR Register ("Registration key two")
$result:= SR Register ("Registration key three")
```

`etc.

SR Document

SR Document(reportArea:L; document:S) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Document	C_STRING(255)	Current document
-> ResultCode	C_LONGINT	Result Code

SR Document returns the current full-pathname for the defined SuperReport Pro area.

ReportArea — The desired SuperReport Pro area.

Document — A valid 4th Dimension variable, which will receive the full-pathname for the defined SuperReport Pro area. If the report area contains a new report or a report that was not read from disk, the *Document* parameter will contain a null string.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will load a report from disk using the **SR Load Report** routine, and then use **the SR Document** routine to determine the full-pathname of the selected file.

```
C_TEXT(tSR_ReportPath)
```

```
C_LONGINT(iSR_Err)
```

```
iSR_Err:=SR Load Report(eReportArea)
```

```
If (iSR_Err=0)
```

```
    iSR_Err:=SR Document(eReportArea;tSR_ReportPath)
```

```
End if
```

SR File Types

SR File Types(platform:L; option:L; value:S) -> resultCode:L

Parameter	Data Type	Description
Platform	C_LONGINT	Desired platform
Option	C_LONGINT	File Type or Creator Option
Value	C_STRING(4)	File Type or Creator (Extension on Windows)
-> ResultCode	C_LONGINT	Result Code

SR File Types allows you to get or set the File Type and/or Creator code for report documents created by SR Save Report, SR Print To Disk, and SR Print HTML commands and their corresponding command when using the SuperReport Pro Editor.

Platform — The desired platform you wish to set the file type or creator.

- 1 – Macintosh platform
- 2 – Windows platform

Action — Desired execution action. You can either get or set the file type and creator codes.

- 1 – Get Creator Code (not applicable on Windows)
- 2 – Get File Type TEXT Code (returns file extension on Windows)
- 4 – Get File Type for SuperReport Pro documents (i.e. SR Save Report)
- 1 – Set the Creator Code (not applicable on Window)
- 2 – Set File Type Code for TEXT files (those created with SR Print To Disk or SR Print HTML)
- 4 – Set File Type Code for SuperReport Pro documents (i.e. SR Save Report)

Value — Corresponding value for defined action. If you are retrieving the value (negative action code), you must pass a valid 4th Dimension variable which will be used to receive the value.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following example will assign the Creator code for SuperReport Pro documents (created using SR Save Report)

```
iSR_Err:=SR File Types(1;1;"SRPD")    `change the default Creator code to SRPD
```

The following routine will get the current Creator code.

```
C_STRING(4;sSR_Creator)
```

```
iSR_Err:=SR File Types(1;-1;sSR_Creator)    `get the creator code
```

SR Get Format Number

SR Get Format Number(formatString:S; fieldType:L) -> resultCode:L

Parameter	Data Type	Description
FormatString	C_STRING	Format String
FieldType	C_LONGINT	Field Type
-> ResultCode	C_LONGINT	Format Number

SR Get Format Number converts a format string to the internal format number. This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

Format String — Desired format string you wish to convert to the internal format number (if it exists).

FieldType — Desired 4D field type of the associated field, which contains the format string. For a complete list of fieldTypes, please refer to the 4th Dimension Language Reference (Type or GET FIELD PROPERTIES commands).

You may choose from one of the following values:

- 4 – Is Date
- 11 – Is Time
- 3 – Is Picture

-> *Result Code* — If the format filter was found, it's associated format number will be returned; otherwise a value of zero (0) will be returned. If you don't use one of the supported fieldTypes, a value of zero (0) will be returned.

SR Get Table List

SR Get Table List(reportArea:L; tableNameList:X; tableNoList:X; options:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report Area
TableNameList	ARRAY STRING	Table name Array
TableNoList	ARRAY INTEGER or ARRAY LONGINT	Tablenumber Array
Options	C_LONGINT	Options
-> ResultCode	C_LONGINT	Format Number

SR Get Table List builds an array of report table names and table numbers (see **SR Structure** for more information).

ReportArea — The desired SuperReport Pro area.

Tablename Array — A valid 4th Dimension string array, which will receive the list of table names for the defined report area.

Tablenumber Array — A valid 4th Dimension integer or longint array, which will receive the list of table numbers for the defined report area.

Note: Tablenumber array is always filled with physical 4D table number.

Options — Options for building table lists. You may choose one of the following values.

0 (SR Structure Physical) – Use physical structure, area is ignored

1 (SR Structure Virtual) – Use virtual structure, area is ignored

2 (SR Structure Area Structure) – Use structure set by SR Structure. If not set, SR Structure Physical | SR Structure Virtual are checked)

4 (SR Structure Get Invisible) – Include invisible tables

8 (SR Structure Get Empty Tables) – Include empty tables (not fields or indexed fields)

16 (SR Structure Sort By Name) – Sort result array by name

256 (SR Structure Mark Indexed) – Returns marked indexed attribute

512 (SR Structure Mark Invisible) – Returns marked invisible attribute

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will fill two 4D arrays with the table name and table number list for the defined report area.

```
ARRAY STRING(32;asSR_Tablenames;0)
```

```
ARRAY LONGINT(aiSR_Tablenumbers;0)
```

```
$SR_Err:=SR Get Table List(eReport;asSR_Tablenames;aiSR_Tablenumbers;SR Structure Area Structure)
```

See Also: SR Get Field List, SR Get Fields As List BLOB, SR Get Tables As List BLOB, SR Get Structure As List BLOB, SR Structure

SR Get Field List

SR Get Field List(reportArea:L; menuID:L; tableNameList:X; tableNoList:X; options:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report Area
MenuID	C_LONGINT	MenuID
FieldnameList	ARRAY STRING	Fieldname Array
FieldnumberList	ARRAY INTEGER or ARRAY LONGINT	Field number Array
FieldTypeList	ARRAY INTEGER or ARRAY LONGINT	Field Type Array
FieldAttribList	ARRAY INTEGER or ARRAY LONGINT	Field Attributes Array
Options	C_LONGINT	Options
-> ResultCode	C_LONGINT	Format Number

SR Get Field List builds an array of report field names and field numbers as well as field types and field attributes (see **SR Structure** for more information).

ReportArea — The desired SuperReport Pro area.

MenuID — Desired menu number whose field information you wish to retrieve.

Fieldname Array — A valid 4th Dimension string array, which will receive the list of field names for the defined report area.

Fieldnumber Array — A valid 4th Dimension integer or longint array, which will receive the list of field numbers for the defined report area.

Note: Fieldnumber array is always filled with physical 4D table number.

FieldTypes Array — A valid 4th Dimension integer or longint array, which will receive the list of field types for the defined report area.

Field Attributes Array — A valid 4th Dimension integer or longint array, which will receive the list of field attributes for the defined report area. For more information on the field attributes, please refer to the 4th Dimension Language Reference (GET FIELD PROPERTIES command).

Options — Options for building table lists. You may choose one of the following values.

0 (SR Structure Physical) – Use physical structure, area is ignored

1 (SR Structure Virtual) – Use virtual structure, area is ignored

2 (SR Structure Area Structure) – Use structure set by SR Structure. If not set, SR Structure Physical | SR Structure Virtual are checked)

4 (SR Structure Get Invisible) – Include invisible tables

8 (SR Structure Get Empty Tables) – Include empty tables (not fields or indexed fields)

16 (SR Structure Sort By Name) – Sort result array by name

256 (SR Structure Mark Indexed) – Returns marked indexed attribute

512 (SR Structure Mark Invisible) – Returns marked invisible attribute

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will fill four 4D arrays with the field information for the defined report area.

```
ARRAY STRING(32;asSR_Fieldnames;0)  
ARRAY LONGINT(aiSR_Fieldnumbers;0)  
ARRAY LONGINT(aiSR_FieldTypes;0)  
ARRAY LONGINT(aiSR_FieldAttributes;0)
```

```
$SR_Err:=SR Get Field List(eReport;3;asSR_ Fieldnames;aiSR_Fieldnumbers; aiSR_FieldTypes;  
aiSR_FieldAttributes;SR Structure Area Structure)
```

See Also: SR Get Table List, SR Get Fields As List BLOB, SR Get Tables As List BLOB, SR Get Structure As List BLOB, SR Structure

SR Get Commands As List BLOB

SR Get Commands As List BLOB(reportArea:L; options:L; resultData:O; excludeThemeList:X; excludeCommandList:X) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report Area
Options	C_LONGINT	Options
Result Data	C_BLOB	Command List BLOB
ExcludeThemeList	ARRAY INTEGER or ARRAY LONGINT	Excluded Theme List
ExcludeCommandList	ARRAY INTEGER or ARRAY LONGINT	Excluded Command List
-> ResultCode	C_LONGINT	Format Number

SR Get Commands As List BLOB builds a 4th Dimension hierarchical list (BLOB) using the reports command list (see SR Commands). You can supply an optional filter array which will remove specific themes and/or specific commands.

This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

ReportArea — The desired SuperReport Pro area.

Options — You may choose one or more of the following values.

- 0 (SR Commands By First Char) – Sort commands by first character of routine name
- 1 (SR Commands By Theme) – Sort commands by theme name
- 2 (SR Commands Area Commands) – Get all commands as configured by SR Commands
- 4 (SR Commands Sort Second Level) – Sort commands by second level
- 8 (SR Commands Sort First Level) – Sort commands by first level

Result Data — A valid 4th Dimension BLOB variable, which will receive the formatted command hierarchical list. You may use this BLOB as a 4th Dimension list object where ever hLists are supported.

ExcludeThemeArray— A 4th Dimension integer or longint array containing the STR# 4 index theme items you wish to exclude from the command list.

ExcludeCommandArray — A 4th Dimension integer or longint array containing the STR# 8 index command items you wish to exclude from the command list.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following will create the command list based on the 4D commands, sorting the list by Theme name.

```
C_BLOB(oSR_Data)
```

```
SET BLOB SIZE(oSR_Data;0)
```

```
$result:=SR Get Commands As List BLOB (eReportArea; SR Commands By Theme;oSR_Data)  
hList:=BLOB To List(oSR_Data) `create a 4D hList object
```

The following example will create the command list based on 4D commands, sorting the list by Theme name and excluding compiler directives.

```
ARRAY LONGINT(aiSR_Theme;1)
```

```
aiSR_Theme{1}:=7 `Compiler
```

```
$result:= SR Get Commands As List BLOB (eReportArea; SR Commands By Theme;  
oSR_Data;"aiSR_Theme")
```

```
hList:=BLOB To List(oSR_Data) `create a 4D hList object
```

The following example will build on the previous example, and include a list of individual routines, which should be omitted (any delete record action) from the command list. The result list will be sorted by Theme name, and each routine will be in alphabetical order within its respective theme.

```
ARRAY LONGINT(aiSR_Theme;1)
```

```
aiSR_Theme{1}:=7 `Compiler
```

```
ARRAY LONGINT(aiSR_Command;3)
```

```
aiSR_Command{1}:=58
```

```
aiSR_Command{2}:=66
```

```
aiSR_Command{3}:=96
```

```
$opts:=SR Commands By Theme | SR Commands Sort First Level | SR Commands Sort Second Level
```

```
$result:= SR Get Commands As List BLOB (eReportArea;$opts;oSR_Data;  
"aiSR_Theme";"aiSR_Command")
```

```
hList:=BLOB To List(oSR_Data) `create a 4D hList object
```

See Also: SR Commands

SR Get Tables As List BLOB

SR Get Tables As List BLOB(reportArea:L; options:L; resultData:O) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report Area
Options	C_LONGINT	Options
Result Data	C_BLOB	Command List BLOB
-> ResultCode	C_LONGINT	Format Number

SR Get Commands As List BLOB builds a 4th Dimension hierarchical list (BLOB) using the table list (see SR Structure).

Item Reference for each item is as follows:

- itemRef in each list item is the table number

This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

ReportArea — The desired SuperReport Pro area.

Options — Options for building table lists. You may choose one of the following values.

0 (SR Structure Physical) – Use physical structure, area is ignored

1 (SR Structure Virtual) – Use virtual structure, area is ignored

2 (SR Structure Area Structure) – Use structure set by SR Structure. If not set, SR Structure Physical | SR Structure Virtual are checked)

4 (SR Structure Get Invisible) – Include invisible tables

8 (SR Structure Get Empty Tables) – Include empty tables (not fields or indexed fields)

16 (SR Structure Sort By Name) – Sort result array by name

Result Data — A valid 4th Dimension BLOB variable, which will receive the formatted table hierarchical list. You may use this BLOB as a 4th Dimension list object where ever hLists are supported.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following will create the table list based on the structure configured using SR Structure. If the routine was not called, the default structure (SR Structure Virtual or SR Structure Physical).

C_BLOB(oSR_Data)

SET BLOB SIZE(oSR_Data;0)

\$result:=SR Get Tables As List BLOB (eReportArea; SR Structure Area Structure;oSR_Data)

hList:=**BLOB To List**(oSR_Data) `create a 4D hList object

See Also: SR Get Field List, SR Get Fields As List BLOB, SR Get Structure As List BLOB, SR Structure

SR Get Fields As List BLOB

SR Get Fields As List BLOB(reportArea:L; tableNo:L; options:L; resultData:O) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report Area
MenuID	C_LONGINT	MenuID
Options	C_LONGINT	Options
Result Data	C_BLOB	Command List BLOB
-> ResultCode	C_LONGINT	Format Number

SR Get Fields As List BLOB builds a 4th Dimension hierarchical list (BLOB) using the fields for a given table (see SR Structure).

Item Reference for each item is as follows:

- itemRef in each list item is the field number

This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

ReportArea — The desired SuperReport Pro area.

MenuID — Desired menu number whose field information you wish to retrieve.

Options — Options for building table lists. You may choose one of the following values.

0 (SR Structure Physical) – Use physical structure, area is ignored

1 (SR Structure Virtual) – Use virtual structure, area is ignored

2 (SR Structure Area Structure) – Use structure set by SR Structure. If not set, SR Structure Physical | SR Structure Virtual are checked)

4 (SR Structure Get Invisible) – Include invisible tables

8 (SR Structure Get Empty Tables) – Include empty tables (not fields or indexed fields)

16 (SR Structure Sort By Name) – Sort result array by name

256 (SR Structure Mark Indexed) – Returns marked indexed attribute

512 (SR Structure Mark Invisible) – Returns marked invisible attribute

Result Data — A valid 4th Dimension BLOB variable, which will receive the formatted field hierarchical list. You may use this BLOB as a 4th Dimension list object where ever hLists are supported.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following will create the field list based on the structure configured using **SR Structure**. If the routine was not called, the default structure (SR Structure Virtual or SR Structure Physical).

C_BLOB(oSR_Data)

C_LONGINT(\$tableNo)

SET BLOB SIZE(oSR_Data;0)

\$tableNo:=**Table**(->[Customer])

\$result:=**SR Get Fields As List BLOB** (eReportArea;\$tableNo; SR Structure Area Structure;oSR_Data)

hList:=**BLOB To List**(oSR_Data) `create a 4D hList object

See Also: SR Get Field List, SR Get Tables As List BLOB, SR Get Structure As List BLOB, SR Structure

SR Get Structure As List BLOB

SR Get Structure As List BLOB(reportArea:L; options:L; resultData:O) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report Area
Options	C_LONGINT	Options
Result Data	C_BLOB	Command List BLOB
-> ResultCode	C_LONGINT	Format Number

SR Get Structure As List BLOB builds a 4th Dimension hierarchical list (BLOB) using the database structure. The hierarchical list will include all tables and their corresponding fields as configured by **SR Structure**.

Item Reference for each item is as follows:

- for tables (root entries), itemRef is the table number
- for fields ([leaf] sublist entries), itemRef is 1000000 + (tableNumber * 1000) + fieldNumber

This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

ReportArea — The desired SuperReport Pro area.

Options — Options for building table lists. You may choose one of the following values.

0 (SR Structure Physical) – Use physical structure, area is ignored

1 (SR Structure Virtual) – Use virtual structure, area is ignored

2 (SR Structure Area Structure) – Use structure set by SR Structure. If not set, SR Structure Physical | SR Structure Virtual are checked)

4 (SR Structure Get Invisible) – Include invisible tables

8 (SR Structure Get Empty Tables) – Include empty tables (not fields or indexed fields)

16 (SR Structure Sort By Name) – Sort result array by name

Result Data — A valid 4th Dimension BLOB variable, which will receive the formatted field hierarchical list. You may use this BLOB as a 4th Dimension list object where ever hLists are supported.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following will create the complete structure list based on the structure configured using **SR Structure**. If the routine was not called, the default structure (SR Structure Virtual or SR Structure Physical).

C_BLOB(oSR_Data)

C_LONGINT(\$tableNo)

SET BLOB SIZE(oSR_Data;0)

\$result:=**SR Get Structure As List BLOB** (eReportArea; SR Structure Area Structure;oSR_Data)

hList:=**BLOB To List**(oSR_Data) `create a 4D hList object

See Also: SR Get Field List, SR Get Table List, SR Get Fields As List BLOB, SR Get Tables As List BLOB, SR Structure

SR Get Variables As List BLOB

SR Get Variables As List BLOB(reportArea:L; options:L; resultData:O) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report Area
Options	C_LONGINT	Options
Result Data	C_BLOB	Command List BLOB
-> ResultCode	C_LONGINT	Format Number

SR Get Variables As List BLOB builds a 4th Dimension hierarchical list (BLOB) using the reports variable list (see SR Variables).

This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

ReportArea — The desired SuperReport Pro area.

Options — You may choose one of the following values.

- 0 – Use alias variable name
- 1 – Use real variable name

Result Data — A valid 4th Dimension BLOB variable, which will receive the formatted variable hierarchical list. You may use this BLOB as a 4th Dimension list object where ever hLists are supported.

The *itemRef* (contained in all list objects) is a valid SuperReport Pro variableID and may be used in routines such as **SR Get Virtual Variable**.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following will create the variable list based on the variables configured using **SR Variables**.

```
C_BLOB(oSR_Data)
```

```
SET BLOB SIZE(oSR_Data;0)
```

```
$result:=SR Get Variables As List BLOB (eReportArea; 0;oSR_Data)  
hList:=BLOB To List(oSR_Data) `create a 4D hList object
```

See Also: SR Variables, SR Get Virtual Variable

SR Get Variable Type

SR Get Variable Type(reportArea:L; varName:S) -> varType:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
VariableName	C_STRING	Variable Name
-> ResultCode	C_LONGINT	Result Code

SR Get Variable Type will return the variable type of a given variable name. The value returned by this routine is identical to that of the 4D command “Type” (with the exception of C_PICTURE variables which will return a value of 10).

This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

ReportArea — The desired SuperReport Pro area.

VarName — Desired report variable name

-> *VarType* — Returns the variable type for the desired variable.

See Also: SR Variables

SR Get Virtual Variable

SR Get Virtual Variable(reportArea::L; varID:L; varName:S; varElement:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
VarID	C_LONGINT	Variable ID
VarName	C_STRING	Variable Name
VarElement	C_LONGINT	Variable Element
-> ResultCode	C_LONGINT	Result Code

SR Get Virtual Variable returns the variable name and variable element associated to an alias variable name created by **SR Variables**.

This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

ReportArea — The desired SuperReport Pro area.

VarID — value from menuID variable returned by call to **SR Power Menu** or refNum from [leaf] list item from **SR Get Variables As List BLOB**.

VarName — Variable Name as set by SR Variables.

VarElement — Variable Element as set by SR Variables.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will obtain the variable name (and variable element where applicable) as configured by **SR Variables** after selecting item with **SR Power Menu**. Upon completion, \$varName will contain the actual 4D variable name and \$varElement would could the direct element should we reference an array.

```
C_LONGINT($menuID)
C_LONGINT($selValue)
C_TEXT($selText;$selAlias)
```

```
C_TEXT($varName)
C_LONGINT($varElement)
```

```
GET MOUSE($mouseX;$mouseY;$mouseButton)
```

```
$menuID:=SR PowerMenu Variables
$ret:=SR Power Menu (vSRArea;$menuID;$mouseX;$mouseY;$selValue;$selText;$selAlias)
$ret:=SR Get Virtual Variable (vSRArea;$menuID;$varName;$varElement)
```

See Also: SR Variables, SR Get Variables As List BLOB, SR Power Menu

SR Get Indexed String

SR Get Indexed String(resID:L; resIndex:L) -> indexedString:S

Parameter	Data Type	Description
ResID	C_LONGINT	STR# ResourceID
ResIndex	C_LONGINT	STR# ResourceIndex
-> ResultCode	C_STRING	Indexed String

SR Get Indexed String is identical to the 4th Dimension command ‘Get Indexed String’ except that it will always use the SuperReport Pro resource file when querying the STR# list.

ResID — Desired STR# resID (must exist in SuperReport Pro resource file)

ResIndex — Desired STR# resIndex (must exist in SuperReport Pro resource file)

-> *IndexedString* — Returns the associated value contained in the STR# resource (resID;resIndex).

Example:

The following example will return the translation text contained in STR# resID-15003; index-32

\$ret:=SR Get Indexed String(15003;32) `returns “At Left Hand Side of Page”

SR Is Modified

SR Is Modified(reportArea::L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
-> ResultCode	C_LONGINT	Result Code

SR Is Modified will return the modified status for the defined SuperReport Pro area. If you wish to procedurally set the modified state for a given SuperReport Pro area, execute the **SR Set Modify** routine.

ReportArea — The desired SuperReport Pro area.

-> *Result Code* — Modified flag for defined SuperReport Pro area.

- 0 – Report area has not been modified
- 1 – Report area has been modified
- 2 – Report area has been modified by report loaded from disk
- 8008 – Invalid area

Example:

The following example will determine if the current SuperReport Pro area has been modified. If it has been modified, it will update the current 4th Dimension report BLOB variable, then set the modified flag to zero.

C_BLOB(pSR_Report)

C_LONGINT(iSR_Err;iSR_Result)

iSR_Result:=**SR Is Modified**(eReportArea)

If (iSR_Result#0) `the report has been modified

 ISR_Err:=**SR Get Area**(eReportData;pSR_Report) `update the report variable

 ISR_Err:=**SR Set Modify**(eReportDat;0)

End if

See Also: SR Set Modified

SR Main Table (formerly SR Main File)

SR Main Table(reportObject:P; action:L; iterations:L; varName:S) -> resultCode:L

Parameter	Data Type	Description
Report Object	C_BLOB	SuperReport Pro Report Object
Action	C_LONGINT	Main File Action
Iterations	C_LONGINT	Report Iterations
Variable Name	C_STRING(32)	4D Variable Name
-> ResultCode	C_LONGINT	Result Code

SR Main Table will set the number of iterations, for a SuperReport Pro report. You can specify the iterations based on the current selection of a 4D table, the size of a 4D array, or a value in a 4D variable.

Report Object — A valid SuperReport Pro Object. This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

Action — Desired configuration action. If you pass a value of zero (0) as the action, you must pass a valid 4th Dimension variable to *Iterations* parameter so that the current value can be returned.

You may choose one of the following values:

- 1 - [SR MainTable Show Dialog](#)
- 0 - [SR MainTable Query](#)
- 1 - [SR MainTable Choose Table](#)
- 2 - [SR MainTable Fixed Iteration](#)
- 4 - [SR MainTable Variable](#)
- 8 - [SR MainTable Array](#)

- 1 – Show Main File/Iterations Dialog. This should only be called when SuperReport Pro Editor is displayed.
- 0 – Query current setting for report
- 1 – Set the iterations based on the table number specified in the iterations parameter.
- 2 – Set the iterations based on the literal value of the iterations parameter
- 4 – Set the iterations based value stored in the variable specified in the varName parameter.
- 8 – Set the iterations based on the size of the array specified in the varName parameter

Iterations — Depending on the value passed in the *Action* parameter, the *Iterations* parameter will may be either a literal value (when using an *Action* value of 1 or 2), or a valid 4th Dimension variable which will receive the current iteration value (when using an *Action* value of 0).

VarName — The name of the variable, which contains the iteration value when passing a value of 4 or 8 as the *Action* value.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Examples:

The following example will set the iterations based on the current selection of the [Customers] table.

```
C_BLOB(pSR_Report)
C_LONGINT(iSR_Err;iSR_Result)

iSR_Err:=SR Main Table(pSR_Report;1;Table(->[Customers]);””)
```

The following example will set the iterations based on the size of the `aReportData` array.

```
C_BLOB(pSR_Report)
C_LONGINT(iSR_Err;iSR_Result)

ARRAY TEXT(aReportData;3)
aReportData{1}:=”item1”
aReportData{2}:=”item2”
aReportData{3}:=”item3”

iSR_Err:=SR Main Table (pSR_Report;8;0;”aReportData”)
```

The following example will return the current iteration value

```
C_BLOB(pSR_Report)
C_LONGINT(iSR_Err;iSR_Result;iSR_Iterations)

iSR_Err:=SR Main Table(pSR_Report;0;iSR_Iterations;””)
```

The following example will create a new report, set the main file to current selection of the [Customer] table and place the new report in the defined SuperReport Pro area.

```
C_BLOB(pSR_Report)
C_LONGINT(iSR_Err;iSR_Result)
C_LONGINT(iSR_MainFile)
```

Case of

:Form event=On Load

```
iSR_MainFile:=Table(->[Customers])
iSR_Err:=SR New Report(pSR_Report)
iSR_Err:=SR Main Table(pSR_Report;1;iSR_MainFile;”aReportData”)
iSR_Err:=SR Set Area(eReportArea;pSR_Report) `copy the new report to the SR Pro area.
```

End case

See **Also:** SR Main Table2

SR Main Table2 (formerly SR Main File2)

SR Main Table2(reportArea:L; action:L; iterations:L; varName:S) -> resultCode:L

Parameter	Data Type	Description
Report Area	C_LONGINT	SuperReport Pro Object
Action	C_LONGINT	Main File Action
Iterations	C_LONGINT	Report Iterations
Variable Name	C_STRING(32)	4D Variable Name
-> ResultCode	C_LONGINT	Result Code

SR Main Table2 will set the number of iterations, for a SuperReport Pro report. You can specify the iterations based on the current selection of a 4D table, the size of a 4D array, or a value in a 4D variable.

This routine is identical to SR Main Table except the first parameter is a SuperReport Pro object instead of the stored object contained in a BLOB variable or field.

Report Object — A valid SuperReport Pro Object. This is the actual SuperReport Pro area not the stored version. If you wish to use the stored version (contained in a BLOB object), use **SR Main Table**.

Action — Desired configuration action. If you pass a value of zero (0) as the action, you must pass a valid 4th Dimension variable to *Iterations* parameter so that the current value can be returned.

You may choose one of the following values:

- 1 - [SR MainTable Show Dialog](#)
- 0 - [SR MainTable Query](#)
- 1 - [SR MainTable Choose Table](#)
- 2 - [SR MainTable Fixed Iteration](#)
- 4 - [SR MainTable Variable](#)
- 8 - [SR MainTable Array](#)

- 1 – Show Main File/Iterations Dialog. This should only be called when SuperReport Pro Editor is displayed.
- 0 – Query current setting for report
- 1 – Set the iterations based on the table number specified in the iterations parameter.
- 2 – Set the iterations based on the literal value of the iterations parameter
- 4 – Set the iterations based value stored in the variable specified in the varName parameter.
- 8 – Set the iterations based on the size of the array specified in the varName parameter

Iterations — Depending on the value passed in the *Action* parameter, the Iterations parameter will may be either a literal value (when using an *Action* value of 1 or 2), or a valid 4th Dimension variable which will receive the current iteration value (when using an *Action* value of 0).

VarName — The name of the variable, which contains the iteration value when passing a value of 4 or 8 as the *Action* value.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

See Also: SR Main Table

SR On Event

SR On Event(reportArea:L; filter:L; callbackProcName:S; postKey:S) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Event Filter	C_LONGINT	Report Event Filter
Callback Procedure	C_STRING(32)	4D Callback Procedure Name
PostKey	C_LONGINT	Postkey Modifier
-> ResultCode	C_LONGINT	Result Code

SR On Event allows you to install a custom callback procedure, which will be executed each time an event occurs while using the SuperReport Pro area.

Warning: You must not destroy the area (close the form or external window) or modify the area in any way, otherwise you may provoke a system crash.

ReportArea — The desired SuperReport Pro area. You **must** pass a valid SuperReport Pro area as the callback routine is installed for the defined SuperReport Pro area only; it is not a global callback for all SuperReport Pro areas.

Filter — This parameter is reserved for future use. You should pass a value of zero (0) to this parameter.

Callback Method — A valid 4th Dimension method which will be executed when events occur within the defined SuperReport Pro area. The callback method must contain the following header

Parameter 1 (\$1) – C_LONGINT: Area Reference (the SuperReport Pro area which raised the event)

Parameter 2 (\$2) – C_LONGINT: Event Type (see list below for a listing of possible events)

Parameter 3 (\$3) – C_LONGINT: Additional Information (see list below for a list of possible values)

Postkey — The postkey parameter allows you to specify which Command Key (on Macintosh) or Control Key (on Windows) sequence should be generated when a SuperReport Pro event occurs.

By placing an invisible button with the same Command key equivalent on layout containing a SuperReport Pro plug-in area, the script of this button will execute whenever a SuperReport Pro event occurs. The postKey parameter is an integer value corresponding to the ASCII character value of the key to be posted. SuperReport Pro automatically posts the key event as a Command Key (on Macintosh) or Control Key (on Windows). To switch off this capability, pass a value of 0 as the postKey parameter.

The postKey event is posted after your callback procedure is executed. This allows you to determine the event type in your callback procedure, set up process or interprocess variables, and then take action in the button script as a result of the posted command key event.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Callback Event Codes

- 11 (SR Zoom Area) – Area zoomed (Parameter 3 contains area reference for zoomed window)
- 12 (SR UnZoom Area)– Zoomed window collapsed (un-zoomed)
- 13 (SR Zoom Area to Back) – Content of the zoomed window was copied back to form area
- 14 (SR Orig Area to Back) – Content of form area was copied to zoomed window
- 15 (SR Area Closing) – Editor window has been closed (either report area on form or external window)
- 20 (SR Editor Mode) – Switched from Editor mode to Preview mode
- 21 (SR Preview Mode) – Switched from Preview mode to Editor mode
- 30 (SR Preview First Page) – In preview, First page option selected
- 31 (SR Preview Previous Page) – In preview, Previous page option selected
- 32 (SR Preview Next Page) – In preview, Next page option selected
- 33 (SR Preview Last Page) – In preview, Last page option selected
- 34 (SR Preview Print Page) – In preview, Print one page option selected
- 35 (SR Close Preview) – In preview, Close preview selected (event 21 can be generated instead)
- 40 (SR Menu Item Selected Before) – Menu Item Selected Before
- 41 (SR Menu Item Selected After) – Menu Item Selected After

Warning: Be careful when dealing with zoomed areas, since a zoomed window, and original areas have **different** references (even if one is a zoomed window of the other). When you receive the “Area Zoomed” event (value of 11), make a note of the zoomed window’s reference number by taking it from the Additional Information (3rd item) parameter.

Example:

The following example is a sample callback method, executing an ALERT call for each event raised by a given SuperReport Pro area. After the callback method is executed, a Command (or Control on Window) -\ event will be executed, triggering an invisible button with the same keyboard equivalent.

```
iSR_Err:=SR On Event(eReportArea;0;" SR_ON_EVENT";Ascii("\"))
```

```
`PM: SR_ON_EVENT
```

```
C_LONGINT($1) `area  
C_LONGINT($2) `event type  
C_LONGINT($3) `addtl info (reserved)  
C_LONGINT(iSR_Event) `event type  
C_TEXT(eventTxt)
```

```
$eventTxt:=""  
iSR_Event:=$2
```

Case of

```
: (iSR_Event=SR Zoom Area )  
  $eventTxt:="Area Zoomed"  
: (iSR_Event=SR UnZoom Area )  
  $eventTxt:="Zoomed Area collapsed"  
: (iSR_Event=SR Zoom Area to Back )  
  $eventTxt:="Zoomed Window brought to front"  
: (iSR_Event=SR Orig Area to Back )  
  $eventTxt:="Original Windows brought to front"  
: (iSR_Event=SR Area Closing )  
  $eventTxt:="SuperReport Pro area closing"  
: (iSR_Event=SR Editor Mode )  
  $eventTxt:="Switching to Preview mode"  
: (iSR_Event=SR Preview Mode )  
  $eventTxt:="Switching to Editor mode"  
: (iSR_Event=SR Preview First Page )  
  $eventTxt:="Preview Button: First Page"  
: (iSR_Event=SR Preview Previous Page )  
  $eventTxt:="Preview Button: Previous Page"  
: (iSR_Event=SR Preview Next Page )  
  $eventTxt:="Preview Button: Next Page"  
: (iSR_Event=SR Preview Last Page )  
  $eventTxt:="Preview Button: Last Page"  
: (iSR_Event=SR Preview Print Page )  
  $eventTxt:="Preview Button: Print One Page"  
: (iSR_Event=SR Close Preview )  
  $eventTxt:="Preview Button: Close Preview"  
: (iSR_Event=SR Menu Item Selected Before )  
  $eventTxt:="Menu Item Selected - Before"  
: (iSR_Event=SR Menu Item Selected After )  
  $eventTxt:="Menu Item Selected - After"
```

End case

```
ALERT($eventTxt)
```

`SC: Button script of Command - \ invisible button

`DC: This is the button script for our button which has the Command - "/" key sequence

If (iSR_Event = 20) `switch from Preview back to editor mode

CANCEL `close the input layout

End if

WARNING: It is very important that you do not try to issue the **CLOSE WINDOW** command when you receive a value of 35 as the event parameter in your callback method. If you do, you will CRASH!!

To get around this problem, you will need to send a message to another process to close the window for you, or to use the new postKey capability to generate an ACCEPT or CANCEL action.

The reason for not executing CLOSE WINDOW is that SuperReport Pro calls your callback procedure reverting execution control to 4D, you kill the window, when 4D returns to the calling procedure (SuperReport Pro), it doesn't exist anymore... hence the crash. By using processes, and passing messages between them, and delaying the killing of the window, you are able to resolve this problem.

SR Power Menu

SR Power Menu(reportArea:L; menuID:L; whereH:L; whereV:L; selectedValue:L; selectedItem:S;
alias:S) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
MenuID	C_LONGINT	Type of menu to display Receives selected menuItemRef
WhereH	C_LONGINT	Horizontal Coordinate
WhereV	C_LONGINT	Vertical Coordinate
SelectedValue	C_LONGINT	Selected Item Value
SelectedItem	C_STRING or C_TEXT	Selected menu item text
SelectedItemAlias	C_STRING or C_TEXT	Selected menu item alias
-> ResultCode	C_LONGINT	Result Code

SR Power Menu provides the ability to display a hierarchical popup menu using a variety of structure objects. When display the popup menu, you can obtain the associated menuID, selectedValue, selectedItem and selectedItemAlias.

This routine is used internally by the SuperReport Pro dialogs and may be used when creating customized report interface. For more information on creating customized report interface, please refer to **Chapter 6 – Command Reference (Object Properties)**.

ReportArea — The desired SuperReport Pro area.

MenuID — When calling the routine, you supply the desired menu structure you wish to display using this parameter. After the routine has completed, this parameter will receive the selected menuItem reference, which can be used by other SuperReport Pro commands such as **SR Get Virtual Variable**.

When calling the routine, you may use one of the following values:

- 0 ([SR PowerMenu Tables](#)) – Displays a popup menu of the table structure (see SR Structure)
- 1 ([SR PowerMenu Fields](#)) – Displays a popup menu of the table/field structure (see SR Structure)
- 2 ([SR PowerMenu Variables](#)) – Displays a popup menu of the variable structure (see SR Variables)
- 3 ([SR PowerMenu Commands](#)) – Displays a popup menu of the command structure (see SR Commands)
- 5 ([SR PowerMenu Break Object Type](#)) – Displays a popup menu of the break objects (see Variable dialog)
- 6 ([SR PowerMenu Variable Type](#)) – Displays a popup menu of the variable types (see Section dialog)
- 7 ([SR PowerMenu Format](#)) – Displays a popup menu of the format filters types (see Field, Variable Dialog)
- 11 ([SR PowerMenu Color](#)) – Display the color popup palette (only works on MacOS)
- 12 ([SR PowerMenu Pattern](#)) – Displays the pattern popup palette (only works on MacOS)
- 13 ([SR PowerMenu Line](#)) – Displays the line thickness popup menu

WhereH — Horizontal coordinate where popup menu should be displayed. If this value (and WhereV) are zero, the popup menu will be positioned at the mouse pointer.

WhereV — Vertical coordinate where popup menu should be displayed. If this value (and WhereH) are zero, the popup menu will be positioned at the mouse pointer.

SelectedItem — A valid 4th Dimension variable which will receive the menuItem position relative to the selected menu. If you are selecting the 3rd item within a submenu, this parameter will contain a value of 3.

SelectedValue — A valid 4th Dimension variable which will receive the selected menuItem text.

SelectedItemAlias — A valid 4th Dimension variable which will receive the selected menuItem alias value when using [SR PowerMenu Tables](#), [SR PowerMenu Fields](#), or [SR PowerMenu Variables](#).

on return, the fields contain:

menuID Type	Selected Item	Selected Value	Alias	menuID
SR PowerMenu Tables	virtual table name	tableID	table alias	
SR PowerMenu Fields	virtual field name	(tableID << 16) fieldID	field alias	
SR PowerMenu Variables	variable name	variable Type	variable alias	value from PopUpMenuSelect (menuID in high word and itemNumber in low word)
SR PowerMenu Commands	menu item string	value from PopUpMenuSelect (menuID in high word and itemNumber in low word)		
SR PowerMenu Break Object Type	menu item string	menu item number - see SR Section Break On		
SR PowerMenu Variable Type	menu item string	menu item number - see SR Variable Type		
SR PowerMenu Format	menu item string	internal format number (counted from zero, for picture formats, it is uses the standard 4D PICT formats, except On Background which is not applicable.		
SR PowerMenu Color	index of the color (0-255) Note: This is a string representation of value (e.g. "116")	RGB color usable in 4D		
SR PowerMenu Pattern		internal pattern number (0-32)		
SR PowerMenu Line	menu item string	menu item number (1-9 where 1=Hair, 2=1 Pt, 3=2 Pt, ...)		

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example:

The following example will obtain the variable name (and variable element where applicable) as configured by **SR Variables** after selecting item with **SR Power Menu**. Upon completion, \$varName will contain the actual 4D variable name and \$varElement would could the direct element should we reference an array.

```
C_LONGINT($menuID)
C_LONGINT($selValue)
C_TEXT($selText;$selAlias)
```

```
C_TEXT($varName)
C_LONGINT($varElement)
```

```
GET MOUSE($mouseX;$mouseY;$mouseButton)
```

```
$menuID:=SR PowerMenu Variables
$ret:=SR Power Menu (vSRArea;$menuID;$mouseX;$mouseY;$selValue;$selText;$selAlias)
$ret:=SR Get Virtual Variable (vSRArea;$menuID;$varName;$varElement)
```

See Also: SR Get Virtual Variable

SR RELATIONS

SR RELATIONS(relateOne:L; relateMany:L)

Parameter	Data Type	Description
Relate One	C_LONGINT	Execute Relate One
Relate Many	C_LONGINT	Execute Relate Many

SR RELATIONS is used to control whether SuperReport Pro generates RELATE MANY and RELATE ONE calls as a report is generated, and whether SuperReport Pro makes use of automatic relations or not.

The relateOne and relateMany parameters may take one of the following values:

- 1 – Return current setting
- 0 – Switch off automatic relations and Relate One/Many calls
- 1 – Switch on automatic relations but do not make Relate One/Many calls
- 2 – Switch on automatic relations and make Relate One/Many calls

By default, SuperReport Pro makes use of automatic relations, and issues a RELATE ONE and RELATE MANY command as each record of the main file is processed. This ensures that related information is always available to the report.

In order to improve speed, you can "switch off" this behavior, and you can do this in the Start script of the report itself, and restore the setting in the End script of the report. Note that this command is GLOBAL and will effect SuperReport Pro throughout your database.

The SR RELATIONS command can be used in the Start and End scripts of your reports so that the effect of using the command is "local" to specific reports only. If you execute the statement, SR RELATIONS (0; 0) then you can still place an explicit RELATE ONE or RELATE MANY command in your report scripts to load specific record selection from other files.

Example

Start script switches off all relations to process data as fast as possible before the report starts to print

```
C_LONGINT(iSR_RelateOne;iSR_RelateMany)
```

```
iSR_RelateOne := -1  
iSR_RelateMany := -1
```

```
SR RELATIONS (iSR_RelateOne; iSR_RelateMany) `get current settings  
SR RELATIONS (0; 0)
```

And the End script restores the previous settings after the report is generated

```
SR RELATIONS (iSR_RelateOne; iSR_RelateMany) `restore previous settings
```

SR Set Modify

SR Set Modify(reportArea:L; status:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Status	C_LONGINT	Modified status
-> ResultCode	C_LONGINT	Result Code

SR Set Modify is used to either set or clear the report changed status for the report in the SuperReport Pro area, area.

ReportArea — The desired SuperReport Pro area.

Status — Modified status.

0 – Set status flag to not modified

1 – Set status flag to modified, needs saving

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will set the modified flag to zero. For additional examples on how to use SR Set Modify, please refer to the **SR Is Modified** example.

```
$result := SR Set Modify (eReportArea; 0) `clear the modify flag for the report in eReportArea
```

See Also: SR Is Modified

SR Get Tips Enabled

SR Get Tips Enabled -> tipState:L

Parameter	Data Type	Description
-> tipState	C_LONGINT	Tool Tips State

SR Get Tips Enabled will return to the tool tip state of SuperReport Pro tool tips.

Note: This routine works in the same fashion as 4Ds built-in tool tips. Enabling or disabling tool tips with this routine will affect **all** application tool tips (including those used by 4D).

-> *Result Code* — Tool tip state

0 – Tool tips are not enabled

1 – Tool tips are enabled

Example

The following example determines if 4Ds tool tips are enabled. If not, they are enabled using the **SR SET TIPS ENABLED** routine.

```
$state := SR Get Tips Enabled `returns the tool tip state
```

```
if ($state=0)
```

```
    SR SET TIPS ENABLED($state)
```

```
End if
```

See Also: SR SET TIPS ENABLED

SR SET TIPS ENABLED

SR SET TIPS ENABLED(toolTipState:L)

Parameter	Data Type	Description
toolTipState	C_LONGINT	SR Pro Tool Tip State

SR SET TIPS ENABLED will enable or disable SuperReport Pro's tool tips.

Note: This routine works in the same fashion as 4Ds built-in tool tips. Enabling or disabling tool tips with this routine will affect **all** application tool tips (including those used by 4D).

State — Tool tip state.

- 0 – Disable tool tips
- 1 – Enable tool tips
- 2 – Toggle current state (if off, tips will be turned on; if on, tips will be turned off)

Example

See **SR Get Tips Enabled** example for an example of using SR SET TIPS ENABLED

See Also: SR Get Tips Enabled

SR SWAP HANDLES

Deprecated v2.9

SR SWAP HANDLES(blob:O;picture:P)

Parameter	Data Type	Description
BLOB Object	C_BLOB	SRP Area in blob object
Picture Object	C_PICTURE	Destination PICT object

This command is obsolete and should no longer be used.

SR SWAP HANDLES provided an interface for converting an SRP report template which had been stored in a 4th Dimension blob field (typically used by autosave interface) into a valid SRP picture object, which could be used by other SuperReport Pro routines requiring a picture variable.

Versions 2.9 and above store SRP report templates into Blob fields. Picture objects are no longer used.

Use the **SR Report To BLOB** command (on next page) to convert SRP objects from picture format to Blobs.

SR Report To BLOB

New v2.9

SR Report To BLOB (picture:P) -> blob:O

Parameter	Data Type	Description
picture	C_PICTURE	Source PICT object
-> blob	C_BLOB	Resulting blob

SR Report To BLOB converts SRP objects stored in pictures (with previous SuperReport Pro versions) to the Blob format.

Versions 2.9 and above store SRP report templates into Blob fields. Picture objects are no longer used.

The *UpdateStruct_Pict2blob* project method in the SRP demonstration database preforms this conversion.

Example

The following example converts all reports stored in the [RM_Reports]RM_PictureField picture field into the [RM_Reports]RM_BlobField blob field for further use with SRP 2.9 and above.

```
READ WRITE([RM_Reports])
ALL RECORDS([RM_Reports])

While (Not(End selection([RM_Reports])))
    [RM_Reports]RM_BlobField:=SR Report To BLOB ([RM_Reports]RM_PictureField)
    SAVE RECORD([RM_Reports])
    NEXT RECORD([RM_Reports])
End while
```

SR Pro – Printing

Each of the following routines provides the ability to procedurally configure and print reports created by SuperReport Pro.

Routines included in this section are:

- ***SR Get Number Of Pages***
- SR Page Setup
- SR Preview
- *SR Print Disk*
- SR Print PICT
- SR Print Report
- *SR Print HTML*
- **SR Print BLOB**
- SR Validate

SR Get Number Of Pages

SR Get Number Of Pages(reportArea:L;printOpts:L;sectionFlag:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_BLOB	Report
PrintOption	C_LONGINT	Print Options
SectionFlag	C_LONGINT	Print Section Flags
-> NumPages	C_LONGINT	Number of pages

SR Get Number Of Pages will return the number of pages for the desired report, using the supply print options and section information.

Report Object — A valid SuperReport Pro report (BLOB format). This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

Print Options — Print options flags. You can pass the sum of the values you wish to execute, using the following table:

- 0 – Default options, don't do anything special
- 1 ([SR PrintToPict Validate](#)) – Validate Page Setup
- 2 ([SR PrintToPict Job Dialog](#)) – Show Job Setup
- 4 ([SR PrintToPict No Progress](#)) – Don't display progress bar

If the Page Setup is not correct for your chosen printer, and you specify to check the Page Setup using the option parameter, then the Page Setup dialog will be displayed. If the user cancels either the Page Setup or Job Setup dialogs then the report will not be printed.

Section Flag — The sections parameter is used to control which sections of the report are printed (assuming that you have not specified the Body section only using the options parameter). To print specific sections, pass the sum of the section numbers that you require which are listed in the **Chapter 7 - SuperReport Pro Codes (Section Codes)**.

You may choose one or more of the following.

- 0 - [SR All Sections](#)
- 1 - [SR Section Header Mask](#)
- 2 - [SR Section SubHeader1 Mask](#)
- 4 - [SR Section SubHeader2 Mask](#)
- 8 - [SR Section SubHeader3 Mask](#)
- 16 - [SR Section SubHeader4 Mask](#)
- 32 - [SR Section SubHeader5 Mask](#)
- 64 - [SR Section SubHeader6 Mask](#)
- 128 - [SR Section Body Mask](#)
- 256 - [SR Section SubTotal6 Mask](#)
- 512 - [SR Section SubTotal5 Mask](#)
- 1024 - [SR Section SubTotal4 Mask](#)
- 2048 - [SR Section SubTotal3 Mask](#)
- 4096 - [SR Section SubTotal2 Mask](#)
- 8192 - [SR Section SubTotal1 Mask](#)
- 16384 - [SR Section Total Mask](#)
- 32768 - [SR Section Footer Mask](#)

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

C_BLOB(pSR_ReportData)

C_LONGINT(iSR_NumPages)

pSR_ReportData := [ReportMgr]ReportData

iSR_NumPages:= ***SR Get Number Of Pages*** (pSR_ReportData;0;SR All Sections)

SR Page Setup

SR Page Setup(reportArea:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
-> ResultCode	C_LONGINT	Result Code

SR Page Setup is used to display the Page Setup for the specified report. You must pass the report parameter as a global variable since the user may modify the Page Setup, and you will need to get the new changes back.

ReportArea — The desired SuperReport Pro area.

-> *Result Code* — If the command returns a value of 1 then the page setup was changed and saved by the user, and you should save the modified report.

Example

C_BLOB(pSR_ReportData)

C_LONGINT(iSR_Result)

```
pSR_ReportData := [ReportMgr]ReportData
iSR_Result:= SR Page Setup (pSR_ReportData)
If (iSR_Result = 1)
  [ReportMgr]ReportData:= pSR_ReportData
End If
```

See Also: SR Validate, SR Preview, SR Print Report, SR Print Disk, SR Print HTML, SR Print To BLOB, SR Print PICT

SR Preview

SR Preview(reportObject:P; left:L; top:L; right:L; bottom:L; windowType:L; windowTitle:S {; modalFlag:L}) -> resultCode:L

Parameter	Data Type	Description
ReportObject	C_BLOB	Report Object
Left	C_LONGINT	Window Position – Left
Top	C_LONGINT	Window Position – Top
Right	C_LONGINT	Window Position – Right
Bottom	C_LONGINT	Window Position – Bottom
Window Type	C_LONGINT	Window Type
Window Title	C_STRING(255)	Window Title
Options	C_LONGINT	Operation Options
-> ResultCode	C_LONGINT	Result Code

SR Preview is used to display a report in an external window in preview mode. When the user hits the "X" (Done) button, the window is automatically closed. You must ensure that the window type you specify has a close box for this feature to work (SuperReport Pro posts a mouse-down event in the window's close box when the "X" button is pressed).

Report Object — A valid SuperReport Pro Object. This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

Left — Left coordinate for preview window.

Top — Top coordinate for preview window.

Right — Right coordinate for preview window.

Bottom — Bottom coordinate for preview window.

Window Type — Window type for preview window. The window type parameter is the same as that used by the 4th Dimension Open window or Open external window routine. For more information on window types, please refer to the **4th Dimension Language Reference**.

Window Title — Window title for preview window.

Options — Bitwise operator to customize the functionality of the preview operation

0 – No options (default)

1 – Perform preview in modal state (user must close window before continuing)

2 – Open window in zoomed state

Passing a value of zero (or omitting the parameter) will display the preview window, and then continue code execution.

Passing a value of one will display the preview window and halt code execution until the preview window is closed.

-> *Result Code* —If the *ModalFlag* is set to 1, the result value will be zero when the window is closed. If the *ModalFlag* is set to 0, the routine will return a valid window reference if the report opened successfully; otherwise a valid SuperReport Pro result code will be returned.

Examples

The following example will display the desired report in a preview window, then continue with code execution.

```
C_BLOB(pSR_ReportData)
```

```
C_LONGINT(iSR_WinRef)
```

```
iSR_WinRef:=SR Preview(pSR_ReportData;50;50;Screen width-50;Screen height-50;8;"SR Pro Preview")
```

The following example will display the desired report in a preview window (modal window).

```
C_BLOB(pSR_ReportData)
```

```
C_LONGINT(iSR_WinRef)
```

```
iSR_WinRef:=SR Preview(pSR_ReportData;50;50;Screen width-50;Screen height-50;8;"SR Pro Preview";1)
```

See Also: SR Page Setup, SR Print Report, SR Print Disk, SR Print HTML, SR Print To BLOB, SR Print PICT

SR Print Disk

SR Print Disk(reportObject:P; documentName:S; printOption:L; sectionFlags:L; fieldDelim:L; recDelim:L; objectOrder:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Document Name	C_STRING(255)	Disk filename
PrintOption	C_LONGINT	Print Options
SectionFlag	C_LONGINT	Print Section Flags
FieldDelim	C_LONGINT	Field Delimiter
RecDelim	C_LONGINT	Record Delimiter
Object Order	C_LONGINT	Object Order
-> ResultCode	C_LONGINT	Result Code

SR Print Disk allows you to print a SuperReport Pro report to disk without requiring a SuperReport Pro area - the output written to the specified document is an ASCII interpretation of the report. Thus SuperReport Pro can be used to create export documents.

NOTE: The Header and Footer sections are only printed once as the report is generated since the concept of pages don't exist when writing the report to a document.

Report Object — A valid SuperReport Pro Object. This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

Document Name — A valid document pathname. If the *Document Name* parameter is a null string, the standard Save File dialog will be displayed. If you want to specify a default value and allow the user to determine where the file will be saved, pass a value as the *Document Name* and use pass a value of one to the Print Options parameter.

Print Options — Print options flags. You can pass the sum of the values you wish to execute, using the following table:

- 0 – Default options, don't do anything special
- 1 (SR PrintToDisk File Dialog) – Display Save File dialog, regardless of Document Name
- 2 (SR PrintToDisk Body Only) – Print Body section only, regardless of sectionFlag
- 4 (SR PrintToDisk Static Text) – Print Static text items

Section Flag — The sections parameter is used to control which sections of the report are printed (assuming that you have not specified the Body section only using the options parameter). To print specific sections, pass the sum of the section numbers that you require which are listed in the **Chapter 7 - SuperReport Pro Codes (Section Codes)**.

You may choose one or more of the following.

- 0 - SR All Sections
- 1 - SR Section Header Mask
- 2 - SR Section SubHeader1 Mask
- 4 - SR Section SubHeader2 Mask
- 8 - SR Section SubHeader3 Mask
- 16 - SR Section SubHeader4 Mask
- 32 - SR Section SubHeader5 Mask
- 64 - SR Section SubHeader6 Mask
- 128 - SR Section Body Mask
- 256 - SR Section SubTotal6 Mask
- 512 - SR Section SubTotal5 Mask
- 1024 - SR Section SubTotal4 Mask
- 2048 - SR Section SubTotal3 Mask
- 4096 - SR Section SubTotal2 Mask
- 8192 - SR Section SubTotal1 Mask
- 16384 - SR Section Total Mask
- 32768 - SR Section Footer Mask

Field Delimiter — The ASCII field delimiter that will be used to separate report objects in the exported file. Generally, this value will be a TAB character (ASCII value 9).

Record Delimiter — The ASCII record delimiter that will be used to separate report records in the exported file. General, this value will be a CR (ASCII 13).

Object Order — When printing to disk, you may determine the order in which objects are printed. You may choose one of the following values:

- 0 (SR Generic Option Set Off) – Use print order (see **SR Reorder Objects** for more information)
- 1 (SR Generic Option Set On) – Use object order (top/left coordinate)

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will print a saved report in a 4th Dimension report record, printing all sections except the header and footer.

```
C_LONGINT(iSR_Err)
```

```
C_LONGINT($options;$section)
```

```
C_STRING(255;sSR_Document)
```

```
$options:=1+4 `display save file dialog, and print static text items
```

```
$section:=65535-1-32768 `print all sections except the Header and Footer
```

```
iSR_Err:=SR Print Disk([ReportMgr]ReportData;sSR_Document;$options;$section;9;13)
```

See Also: SR Page Setup SR Preview, SR Print Report, SR Print HTML, SR Print To BLOB, SR Print PICT

SR Print PICT

SR Print PICT(reportObject:P; pictArrayName:S; options:L; sections:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_BLOB	Report
PictArrayName	C_STRING(32)	Name of PICT array
PrintOption	C_LONGINT	Print Options
SectionFlag	C_LONGINT	Print Section Flags
-> ResultCode	C_LONGINT	Result Code

SR Print PICT allows you to print a SuperReport Pro report to a picture array without requiring a SuperReport Pro area - the printed pages are placed in each element of the picture array. This command can be used to implement your own print preview or alternate web printing system (instead of using SR Print HTML).

Report Object — A valid SuperReport Pro report (BLOB format). This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

Array Name — Name of a valid 4th Dimension PICT array, which will receive the pages of the printed report.

Print Options — Print options flags. You can pass the sum of the values you wish to execute, using the following table:

- 0 – Default options, don't do anything special
- 1 ([SR PrintToPict Validate](#)) – Validate Page Setup
- 2 ([SR PrintToPict Job Dialog](#)) – Show Job Setup
- 4 ([SR PrintToPict No Progress](#)) – Don't display progress bar

If the Page Setup is not correct for your chosen printer, and you specify to check the Page Setup using the option parameter, then the Page Setup dialog will be displayed. If the user cancels either the Page Setup or Job Setup dialogs then the report will not be printed.

Section Flag — The sections parameter is used to control which sections of the report are printed (assuming that you have not specified the Body section only using the options parameter). To print specific sections, pass the sum of the section numbers that you require which are listed in the **Chapter 7 - SuperReport Pro Codes (Section Codes)**.

You may choose one or more of the following.

- 0 - [SR All Sections](#)
- 1 - [SR Section Header Mask](#)
- 2 - [SR Section SubHeader1 Mask](#)
- 4 - [SR Section SubHeader2 Mask](#)
- 8 - [SR Section SubHeader3 Mask](#)
- 16 - [SR Section SubHeader4 Mask](#)
- 32 - [SR Section SubHeader5 Mask](#)
- 64 - [SR Section SubHeader6 Mask](#)
- 128 - [SR Section Body Mask](#)
- 256 - [SR Section SubTotal6 Mask](#)
- 512 - [SR Section SubTotal5 Mask](#)
- 1024 - [SR Section SubTotal4 Mask](#)
- 2048 - [SR Section SubTotal3 Mask](#)
- 4096 - [SR Section SubTotal2 Mask](#)
- 8192 - [SR Section SubTotal1 Mask](#)
- 16384 - [SR Section Total Mask](#)
- 32768 - [SR Section Footer Mask](#)

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will create a 4D PICT array, one element for each page of the report.

```
C_LONGINT(iSR_Result;$options;$section)
```

```
ARRAY PICTURE(apSR_Report;0)
```

```
$options := 1 + 2 `valid Page Setup, show Job Setup dialog
```

```
$section := 0 `all sections
```

```
iSR_Result:= SR Print PICT ([ReportMgr]ReportData; "apSR_Report"; $options; $section)
```

See Also: SR Page Setup SR Preview, SR Print Report, SR Print Disk, SR Print HTML, SR Print To BLOB

SR Print Report

SR Print Report(reportObject:P; options:L; sections:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
PrintOption	C_LONGINT	Print Options
SectionFlag	C_LONGINT	Print Section Flags
-> ResultCode	C_LONGINT	Result Code

SR Print Report allows you to print a SuperReport Pro report without requiring a SuperReport Pro area.

Report Object — A valid SuperReport Pro Object. This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

Print Options — Print options flags. You can pass the sum of the values you wish to execute, using the following table:

- 0 – Default options, don't do anything special
- 1 ([SR Print Option Validate](#)) – Validate Page Setup
- 2 ([SR Print Option Job Dialog](#)) – Show Job Setup
- 4 ([SR Print Option No Progress](#)) – Don't display progress bar

If the Page Setup is not correct for your chosen printer, and you specify to check the Page Setup using the option parameter, then the Page Setup dialog will be displayed. If the user cancels either the Page Setup or Job Setup dialogs then the report will not be printed.

Section Flag — The sections parameter is used to control which sections of the report are printed (assuming that you have not specified the Body section only using the options parameter). To print specific sections, pass the sum of the section numbers that you require which are listed in the **Chapter 7 - SuperReport Pro Codes (Section Codes)**.

You may choose one or more of the following.

- 0 - [SR All Sections](#)
- 1 - [SR Section Header Mask](#)
- 2 - [SR Section SubHeader1 Mask](#)
- 4 - [SR Section SubHeader2 Mask](#)
- 8 - [SR Section SubHeader3 Mask](#)
- 16 - [SR Section SubHeader4 Mask](#)
- 32 - [SR Section SubHeader5 Mask](#)
- 64 - [SR Section SubHeader6 Mask](#)
- 128 - [SR Section Body Mask](#)
- 256 - [SR Section SubTotal6 Mask](#)
- 512 - [SR Section SubTotal5 Mask](#)
- 1024 - [SR Section SubTotal4 Mask](#)
- 2048 - [SR Section SubTotal3 Mask](#)
- 4096 - [SR Section SubTotal2 Mask](#)
- 8192 - [SR Section SubTotal1 Mask](#)
- 16384 - [SR Section Total Mask](#)
- 32768 - [SR Section Footer Mask](#)

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will print the report stored in the users database, printing only the body section.

```
C_LONGINT(iSR_Result;$options;$section)
```

```
ARRAY PICTURE(apSR_Report;0)
```

```
$options := 1 + 2 `valid Page Setup, show Job Setup dialog
```

```
$section := 128 `body section only
```

```
iSR_Result:= SR Print Report ([ReportMgr]ReportData; $options; $section)
```

See Also: SR Page Setup SR Preview, SR Print Disk, SR Print HTML, SR Print To BLOB, SR Print PICT

SR Print HTML

SR Print HTML(reportObject:P; documentName:S; options:L; sectionFlag:L; objectOrder:L) -> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
DocumentName	C_STRING(32)	Name of HTML Document
PrintOption	C_LONGINT	Print Options
SectionFlag	C_LONGINT	Print Section Flags
Object Order	C_LONGINT	Print Object Order
-> ResultCode	C_LONGINT	Result Code

SR Print HTML allows you to print the defined SuperReport Pro report in HTML format.. All HTML tags defined using HTML dialog for the various report objects will be used. All font attribute settings will be used based on their definition in the report, unless a tag has been supplied.

Report Object — A valid SuperReport Pro Object. This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

Document Name — Full-pathname or local pathname of the HTML document you wish to create. If you pass a null string, the standard Save File dialog will be displayed. If you want to specify a default value and allow the user to determine where the file will be saved, pass a value as the *Document Name* and use pass a value of one to the Print Options parameter.

Print Options — Print options flags. You can pass the sum of the values you wish to execute, using the following table:

- 0 – Default options, don't do anything special
- 1 ([SR PrintToDisk File Dialog](#)) – Display Save File dialog, regardless of Document Name
- 2 ([SR PrintToDisk Body Only](#)) – Print Body section only, regardless of sectionFlag
- 4 ([SR PrintToDisk Static Text](#)) – Print Static text items

Section Flag — The sections parameter is used to control which sections of the report are printed (assuming that you have not specified the Body section only using the options parameter). To print specific sections, pass the sum of the section numbers that you require which are listed in the **Chapter 7 - SuperReport Pro Codes (Section Codes)**.

You may choose one or more of the following.

- 0 - [SR All Sections](#)
- 1 - [SR Section Header Mask](#)
- 2 - [SR Section SubHeader1 Mask](#)
- 4 - [SR Section SubHeader2 Mask](#)
- 8 - [SR Section SubHeader3 Mask](#)
- 16 - [SR Section SubHeader4 Mask](#)
- 32 - [SR Section SubHeader5 Mask](#)
- 64 - [SR Section SubHeader6 Mask](#)
- 128 - [SR Section Body Mask](#)
- 256 - [SR Section SubTotal6 Mask](#)
- 512 - [SR Section SubTotal5 Mask](#)
- 1024 - [SR Section SubTotal4 Mask](#)
- 2048 - [SR Section SubTotal3 Mask](#)
- 4096 - [SR Section SubTotal2 Mask](#)
- 8192 - [SR Section SubTotal1 Mask](#)
- 16384 - [SR Section Total Mask](#)
- 32768 - [SR Section Footer Mask](#)

Object Order— When printing to disk, you may determine the order in which objects are printed. You may choose one of the following values:

0 (SR Generic Option Set Off) – Use print order (see **SR Reorder Objects** for more information)

1 (SR Generic Option Set On) – Use object order (top/left coordinate)

-> *Result Code*—If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will set the default filename for the HTML file to “myReport.html”, display the Save File dialog, and print all sections.

C_LONGINT(iSR_Result;\$options;\$section)

C_TEXT(tSR_Document)

tSR_Document:=”myReport.html”

\$options := 1 `show Save File dialog

\$section := 0 `all sections

iSR_Result:= **SR Print HTML** ([ReportMgr]ReportData; tSR_Document; \$options; \$section;1)

See Also: SR Page Setup SR Preview, SR Print Report, SR Print Disk, SR Print To BLOB, SR Print PICT

SR Print To BLOB

SR Print To BLOB(reportObject:P; reportData:O; options:L; sectionFlag:L; fieldDelim:L; recDelim:L; objectOrder:L)
-> resultCode:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Report Data	C_BLOB	4D BLOB Variable
PrintOption	C_LONGINT	Print Options
SectionFlag	C_LONGINT	Print Section Flags
FieldDelim	C_LONGINT	Field Delimiter
RecDelim	C_LONGINT	Record Delimiter
Object Order	C_LONGINT	Printed Object Order
-> ResultCode	C_LONGINT	Result Code

SR Print To BLOB allows you to print the defined SuperReport Pro report to a 4th Dimension BLOB variable. This is very useful when you wish to display reports created by SuperReport Pro as part of a dynamic web page using a tool such as Active4D.

All HTML tags defined using HTML dialog for the various report objects will be used. All font attribute settings will be used based on their definition in the report, unless a tag has been supplied.

Report Object — A valid SuperReport Pro Object. This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

Report Data — A valid 4th Dimension BLOB variable, which will receive the report output.

Print Options — Print options flags. You can pass the sum of the values you wish to execute, using the following table:

- 1 ([SR PrintToBLOB Output HTML](#)) – Print using HTML attributes, otherwise standard text output
- 2 ([SR PrintToBLOB Body Only](#)) – Only output the Body section
- 3 ([SR PrintToBLOB Static Text](#)) – Include static text items
- 4 ([SR PrintToBLOB TopLeft Order](#)) – Print using top/left coordinates, otherwise print order will be used

Section Flag — The sections parameter is used to control which sections of the report are printed (assuming that you have not specified the Body section only using the options parameter). To print specific sections, pass the sum of the section numbers that you require which are listed in the **Chapter 7 - SuperReport Pro Codes (Section Codes)**.

You may choose one or more of the following.

- 1 - [SR Section Header Mask](#)
- 2 - [SR Section SubHeader1 Mask](#)
- 4 - [SR Section SubHeader2 Mask](#)
- 8 - [SR Section SubHeader3 Mask](#)
- 16 - [SR Section SubHeader4 Mask](#)
- 32 - [SR Section SubHeader5 Mask](#)
- 64 - [SR Section SubHeader6 Mask](#)
- 128 - [SR Section Body Mask](#)
- 256 - [SR Section SubTotal6 Mask](#)
- 512 - [SR Section SubTotal5 Mask](#)
- 1024 - [SR Section SubTotal4 Mask](#)
- 2048 - [SR Section SubTotal3 Mask](#)
- 4096 - [SR Section SubTotal2 Mask](#)
- 8192 - [SR Section SubTotal1 Mask](#)
- 16384 - [SR Section Total Mask](#)
- 32768 - [SR Section Footer Mask](#)

65535 - SR All Sections

Field Delimiter — The ASCII field delimiter that will be used to separate report objects in the exported file. Generally, this value will be a TAB character (ASCII value 9). If you are outputting in HTML format (SR PrintTOBLOB Output HTML), the field delimiter will not be used.

Record Delimiter — The ASCII record delimiter that will be used to separate report records in the exported file. Generally, this value will be a CR (ASCII 13). If you are outputting in HTML format (SR PrintTOBLOB Output HTML), the field delimiter will not be used.

Object Order — When printing to disk, you may determine the order in which objects are printed. You may choose one of the following values:

0 (SR Generic Option Set Off) – Use print order (see **SR Reorder Objects** for more information)

1 (SR Generic Option Set On) – Use object order (top/left coordinate)

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will print the report (ASCII format) to a 4th Dimension BLOB variable, utilizing the SuperReport Pro HTML processor.

```
C_BLOB(oSR_Data)
```

```
C_BLOB($report)
```

```
C_TEXT($text)
```

```
C_LONGINT($options;$section)
```

```
SET BLOB SIZE(oSR_Data;0) `clear the blob
```

```
$options:=SR PrintToBLOB Output HTML | SR PrintToBLOB Static Text | SR PrintToBLOB TopLeft Order
```

```
$section:=SR All Sections
```

```
iSR_Result:=SR Get Area (vSRArea;$report)
```

```
iSR_Result:=SR Print To BLOB ($report;oSR_Data;$options;$section;9;13)
```

```
$text:=BLOB to text(oSR_Data;Text without length )
```

```
$hDoc:=Create document("customerList.html")
```

```
If (OK=1)
```

```
    SEND PACKET($hDoc;$text)
```

```
    CLOSE DOCUMENT($hDoc)
```

```
End if
```

See **Also:** SR Page Setup SR Preview, SR Print Report, SR Print Disk, SR Print HTML, SR Print PICT

SR Validate

SR Page Setup(reportObject:P) -> resultCode:L

Parameter	Data Type	Description
ReportObject	C_BLOB	Report Object
-> ResultCode	C_LONGINT	Result Code

SR Validate will check whether the page setup for the specified report matches the currently selected printer. If not, then the page setup will be modified to match the chosen printer.

Report Object — A valid SuperReport Pro Object. This is not the actual SuperReport Pro area but rather the report object as it would be stored in a 4th Dimension BLOB field or variable.

-> *Result Code* — The validation routine will return one of the following values.

- 0 – Page setup does not match current printer, and has been modified
- 1 – Page setup is correct for current printer
- <0 – An error occurred, a valid SuperReport Pro Error Code will be returned

NOTE: If a value of zero (0) is returned, you should display the Page Setup dialog using the SR Page Setup routine.

Example:

The following example will validate the Page Setup information. If it is not correct (a value of zero is returned), the SR Page Setup routine is called.

```
C_LONGINT(iSR_Result;iSR_Err)
C_BLOB(pSR_ReportData)
```

```
pSR_ReportData := [Report]Format      `make a copy of the report, we might modify it
If (SR Validate (pSR_ReportData) = 0)
iSR_Result:= SR Page Setup (pSR_ReportData) `update page setup data
  If (iSR_Result = 1)
    [ReportMgr]ReportData := pSR_ReportData `stored modified report in [ReportMgr] table
  End If
End If
```

See Also: SR Page Setup

SR Pro – Developer Interface (API)

In addition to the internal forms available when using the default editor, SuperReport Pro provides a complete interface whereby application developers can override the default dialogs, and use custom interface elements and dialogs for most editor events.

Routines included in this section are:

- **SR Set Editor Callback**
- **SR Set Script Callback**
- **SR Get Object IDs**
- **SR Change Object Order**
- **SR Reorder Objects**
- **SR Create Object**
- **SR Delete Object**
- **SR Get/Set Guides**
- **SR Get/Set Object Data**
- **SR Get/Set Object Format**
- **SR Get/Set Object Properties**
- **SR Get/Set Object Scripts**
- **SR Get/Set Scripts**
- **SR Get/Set Sections**
- **SR Get/Set Section Properties**
- **SR Get/Set Section Scripts**
- **SR Get/Set Section Scripts**
- **SR Get/Set Tool**

SR Set Editor Callback

SR Set Editor Callback(reportArea:L; callbackMethod:S) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Callback Method	C_STRING	Report Editor Callback Method
-> ResultCode	C_LONGINT	Result Code

SR Set Editor Callback installs the desired callback method you wish to be executed when a SuperReport Pro Editor event occurs

ReportArea — The desired SuperReport Pro area.

Callback Method — A valid 4th Dimension method which will be executed whenever an editor event is executed.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

The callback method receives the following events:

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area that activated the event
Action Type	C_LONGINT	Report Editor Action
Item Reference	C_LONGINT	Report item associated with event*
Object Type	C_LONGINT	Report Object Type

Item Reference Details:

- If you have activated an object, the value returned is the internal ObjectID.
- If you have activated a section, the value returned is one of the valid Section Objects.
- If you have activated a database script, the value returned is one of ([SR Start Report Script](#), [SR Body Report Script](#), or [SR End Report Script](#))

Report Area — SuperReport Pro area, which raised the event.

Action Type — SuperReport Pro action that was performed. For a complete list of Editor Callback Actions, please refer to **Chapter 7 – SuperReport Pro Codes** (Editor Callback Action Type).

Item Reference — Desired editor item, which raised the event. This value will contain a random (unique) number when a new object is created. If you are editing an existing object, this value will contain the internal object item ID.

Object Type — SuperReport Pro object type. For a complete list of Editor Callback Object Types, please refer to **Chapter 7 – SuperReport Pro Codes** (Editor Callback Object Codes).

-> *Result Code* — The return value will determine how SuperReport Pro will respond after your custom editor code has been executed.

0 – Return a value of zero if you have handled the editor event

1 – Return a value of one if you wish to have SuperReport Pro handle the event (the same action as if the callback method was not installed).

Example:

The following example will install the 4th Dimension method (SR_EditorCallback), which will be executed whenever a SuperReport Pro editor event is sent.

Case of

:(Form event=On Load)
\$SR_Err:=*SR Set Editor Callback*(eReport;"SR_EditorCallback")

End case

Sample Callback Method

The following is a sample callback method, which could be used when installing an editor callback method.

`PM: SR_EditorCallback(area:L; actionType:L; itemRef:L; objType:L)
`LM: 12/3/00, v2.8

`\$1: SR Pro area reference
`\$2: action type
`\$3: object item reference
`\$4: object type

C_LONGINT(\$0) `callback result
C_LONGINT(\$1) `report area
C_LONGINT(\$2) `action type
C_LONGINT(\$3) `object item reference
C_LONGINT(\$4) `object type

\$SR_AREA:=\$1
\$SR_ACTION:=\$2
\$SR_ITEM:=\$3
\$SR_OBJTYPE:=\$4

Case of

:(\$SR_ACTION=SR Editor Create Object) `1
:(\$SR_ACTION=SR Editor Modify Object) `2
:(\$SR_ACTION=SR Editor Modify Section) `3
:(\$SR_ACTION=SR Editor Modify Object Script) `4
:(\$SR_ACTION=SR Editor Modify Report Script) `5
:(\$SR_ACTION=SR Editor Control Click Object) `6
:(\$SR_ACTION=SR Editor Click Object) `7
:(\$SR_ACTION=SR Editor Selection Changed) `8

End case

\$0:=1

SR Set Script Callback

SR Set Script Callback(reportArea:L; callbackMethod:S) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Callback Method	C_STRING	Report Editor Callback Method
-> ResultCode	C_LONGINT	Result Code

SR Set Script Callback installs a callback method that will be used in place of the default script editor for processing of report scripts. This interface provides application developers with an interface for processing scripts which are not possible using standard EXECUTE style interface.

For example, if you are using an enhanced scripting control system such as FootRunner, you create more complex scripts, including complete conditional support and pass that script along to FootRunner at print time for processing.

ReportArea — The desired SuperReport Pro area. Passing a value of zero (0) will install the callback method for all report processing.

Callback Method — A valid 4th Dimension method which will be executed whenever an area script is executed.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

-> *Result Code* — The return value will determine how SuperReport Pro will respond after your custom editor code has been executed.

0 – Return a value of zero if you have handled the editor event

1 – Return a value of one if you wish to have SuperReport Pro handle the event (the same action as if the callback method was not installed).

Example:

The following example will install the 4th Dimension method (SR_ScriptCallback), which will be executed whenever a SuperReport Pro script is called.

```
$SR_Err:=SR Set Editor Callback(0;"SR_EditorCallback") `install the script globally
```

Sample Callback Method

The following is a sample callback method, which could be used when installing an editor callback method.

```
`PM: SR_EXECUTE(scriptData:T)
`LM: 06/17/02, mse

`Invoked:
`
`$ret:=SR Set Script Callback(eSRArea;"SR_EXECUTE")
```

```
C_TEXT($1;$SR_SCRIPT)
```

```
$SR_SCRIPT:=$1
$SR_CMD:=""
```

```
$bUseFoot:=True `set to false if you don't want to process code with FootRunner
```

```
If ($bUseFoot)
  $ret:=FRAppendChecksum ($SR_SCRIPT)
  $ret:=FRRunText ($SR_SCRIPT)
```

```
Else
```

```
  `if you are going to simply pass to 4Ds EXECUTE command, you must parse the
  `script into separate lines as the EXECUTE command requires a single command
```

```
ARRAY TEXT(atSR_ScriptCmd;0)
TC_Parse ($SR_SCRIPT;Char(13);->atSR_ScriptCmd)
For ($i;1;Size of array(atSR_ScriptCmd))
  $SR_CMD:=atSR_ScriptCmd{$i}
  EXECUTE($SR_CMD)
End for
```

```
End if
```

Sample Enhanced Script

When using a custom script executor such as above, your reports can include complex scripts, such as the following. In this script, we are conditionally setting the appears of the [Invoice]Total object based on the current value.

In addition, you will notice that we are calling some customized routines, which are written in 4D to provide simple wrapper routines for developers and users so they aren't required to know the internal working of the SRP Developer API. For more information about creating SRP wrapper routines, please review the SRP Developer API section.

```
If ([Invoice]Total<0)
  SR_SetFont(SRObjctID;"";-1;1)
  SR_SetForeColor(SRObjctID;0x00FF0000)
Else
  SR_SetFont(SRObjctID;"";-1;0)
  SR_SetForeColor(SRObjctID;0x00000000)
End if
```

SR Count Objects

SR Count Objects (reportArea:L; selectedOnlyFlag:L) -> numberOfObjects:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Selected Only Flag	C_LONGINT	Selected Objects Only Flag
-> ResultCode	C_LONGINT	Number of Objects

SR Count Objects returns the number of objects for the desired report area. You may choose to only return the number of selected objects by using the Selected Only Flag.

ReportArea — The desired SuperReport Pro area.

Selected Only Flag — Determines if the array should contain all report objects or only those which are currently selected.

0 - [SR All Objects](#)

1 - [SR Selected Objects](#)

-> *Result Code* —Returns the number of objects for the desired area (greater than or equal to zero); otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will return the total number of objects for the desired report area.

```
$SR_Err:=SR Count Objects($SR_AREA;SR All Objects)
```

See Also: SR Get Object Properties, SR Get Object IDs, SR Set Object Properties, SR Change Object Order, SR Reorder Objects.

SR Get Object IDs

SR Get Object IDs (reportArea:L; selectedOnlyFlag:L; objectIDlist:X) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Selected Only Flag	C_LONGINT	Selected Objects Only Flag
Object ID Array	ARRAY INTEGER or LONGINT	Array of Object IDs
-> ResultCode	C_LONGINT	Result Code

SR Get Object IDs builds a 4th Dimension array containing a list of report object IDs. Using the Selected Only Flag, you can conditionally return only those elements, which are selected, or all report objects.

ReportArea — The desired SuperReport Pro area.

Selected Only Flag — Determines if the array should contain all report objects or only those which are currently selected.

0 - [SR All Objects](#)

1 - [SR Selected Objects](#)

Object List — A valid 4th Dimension array which will receive the list of object IDs.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will build an array of all selected objects for the desired report area.

```
ARRAY INTEGER(aiSR_ObjectList;0)
```

```
$SR_Err:=SR Get Object IDs($SR_AREA;SR Selected Objects;aiSR_ObjectList)
```

See Also: SR Count Objects, SR Get Object Properties, SR Set Object Properties, SR Change Object Order, SR Reorder Objects.

SR Change Object Order

SR Change Object Order (reportArea:L; itemID:L; position:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Item ID	C_LONGINT	Report Object ID
Position	C_LONGINT	Object Position
-> ResultCode	C_LONGINT	Result Code

SR Change Object Order provides the ability to procedurally change the print order of a given report object. You can place an object at the beginning or end of the object list, or move an object forward or backwards.

Note: When reordering an object, it does not physically move the object on the report (unless you move forward/backward); it only moves it within the internal list of objects. This routine is useful when you wish to customize the order in which objects are printed to disk (or HTML files)

ReportArea — The desired SuperReport Pro area.

Object Item ID — A valid report object ID

Object Position — Desired position where you would like to place the object. You may use one of the following SuperReport Pro Constants

- 1 - SR Position At Front
- 2 - SR Position At End
- 3 - SR Position Forward
- 4 - SR Position Backward

When choosing SR Position Forward or SR Position Backward, the object will be moved forward or backwards one position.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example (which is executed in a callback method) will move the object to the end of the object list.

```
$SR_Err:=SR Change Object($SR_AREA;$SR_ITEM;SR Position At End)
```

See Also: SR Count Objects, SR Get Object Properties, SR Get Object IDs, SR Set Object Properties, SR Reorder Objects.

SR Reorder Objects

SR Reorder Objects (reportArea:L; option:L; objectListArray:X) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Move To Front	C_LONGINT	Move to beginning of list
Object List	ARRAY INTEGER or LONGINT	Array of object IDs
-> ResultCode	C_LONGINT	Result Code

SR Reorder Objects provides the ability to procedurally reorder the printing position all objects. Unlike the SR Change Object Order routine, which works on a single object, this routine enables you to provide an ordered list of where you wish to have objects ordered.

Note: When moving an object, it does not physically move the object on the report (unless you move forward/backward); it only moves it within the internal list of objects. This routine is useful when you wish to customize the order in which objects are printed to disk (or HTML files)

ReportArea — The desired SuperReport Pro area.

Move To Front — Move object items to front

0 - SR Generic Option Set Off

1 - SR Generic Option Set On

Object ID List — A valid 4th Dimension integer or longint array containing object IDs you wish to move. If you would like to build an array of SuperReport Pro objects, use the **SR Get Object IDs** routine.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example moves the array of objects to front of object list.

```
$SR_Err:=SR Reorder Objects($SR_AREA; SR Generic Option Set Off;aiSR_ObjectID)
```

See Also: SR Count Objects, SR Get Object Properties, SR Get Object IDs, SR Set Object Properties, SR Change Object Order.

SR Create Object

SR Create Object (reportArea:L; objName:S; recTop:L; rectLeft:L; rectBottom:L; rectRight:L; objType:L; options:L; selected:L; tableNo:L; fieldNo:L; varType:L; arrayElement:L; calcType:L; calcName:S; rows:L; cols:L; repeatHOffset:L; repeatVOffset:L) -> itemID/result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
ObjName	C_STRING or C_TEXT	Object Name
RectTop	C_LONGINT	Object Position – Top
RectLeft	C_LONGINT	Object Position – Left
RectBottom	C_LONGINT	Object Position – Bottom
RectRight	C_LONGINT	Object Position – Right
ObjType	C_LONGINT	Object Type
Options	C_LONGINT	Object Options
Selected	C_LONGINT	Object Selected
TableNo	C_LONGINT	Object Table Number
FieldNo	C_LONGINT	Object Field Number
VarType	C_LONGINT	Object Variable Type
ArrayElem	C_LONGINT	Array Element
CalcType	C_LONGINT	Object Calc Type
CalcName	C_STRING or C_TEXT	Object Calc Name
Rows	C_LONGINT	Object Rows
Cols	C_LONGINT	Object Cols
RepeatHOffset	C_LONGINT	Object Horizontal Repeat Offset
RepeatVOffset	C_LONGINT	Object Vertical Repeat Offset
-> ResultCode	C_LONGINT	New Item ID or Error Code

SR Create Object provides the ability to procedurally create new objects. You can create any type of report object supported by SuperReport Pro and may define all elements (including object position, calculation options, etc.)

ReportArea — The desired SuperReport Pro area.

Object Name — Desired object name. If you are creating an object, which provides static information (i.e. a static text object or variable), you can supply the information in the Object Name parameter.

RectTop — Object Position – Top. The object position is relative to the upper left corner of the report area.

RectLeft — Object Position – Left. The object position is relative to the upper left corner of the report area.

RectBottom — Object Position – Bottom. The object position is relative to the upper left corner of the report area.

RectRight — Object Position – Right. The object position is relative to the upper left corner of the report area.

ObjType — Desired type of object you are creating. You may use one of the following values.

- 1 - [SR Object Type Text](#)
- 2 - [SR Object Type Field](#)
- 3 - [SR Object Type Variable](#)
- 4 - [SR Object Type Line](#)
- 5 - [SR Object Type Rectangle](#)
- 6 - [SR Object Type Circle](#)
- 8 - [SR Object Type Picture](#)

If you are creating a static text object ([SR Object Type Text](#)) or picture object ([SR Object Type Picture](#)) you can set the static information using **SR Set Object Data** after this routine has completed.

Options — Object Options. The value passed to this routine will vary depending on the type of object you are creating. You may use one or more of the following values. When using multiple values, pass the sum of all objects you wish to activate.

- 1 - [SR Obj Flag Left Line](#)
- 2 - [SR Obj Flag Top Line](#)
- 4 - [SR Obj Flag Right Line](#)
- 8 - [SR Obj Flag Bottom Line](#)
- 15 - [SR Obj Flag All Lines](#)
- 16 - [SR Obj Flag Fixed Horizontal](#)
- 32 - [SR Obj Flag Fixed Vertical](#)
- 64 - [SR Obj Flag Grow Horizontal](#)
- 128 - [SR Obj Flag Grow Vertical](#)
- 256 - [SR Obj Flag Variable Width](#)
- 512 - [SR Obj Flag Variable Height](#)
- 1024 - [SR Obj Flag Replace If Empty](#)
- 2048 - [SR Obj Flag Use Calc Value](#)
- 4096 - [SR Obj Flag Show Calc Value](#)
- 8192 - [SR Obj Flag Repeating Object](#)
- 16384 - [SR Obj Flag Repeat Vertically](#)
- 32768 - [SR Obj Flag Repeat Horizontally](#)
- 65536 - [SR Obj Flag Replace Row If Empty](#)

Selected — Determines if the object will be selected after it appears on the report.

- 0 – Object is not selected
- 1 – Object is selected

TableNo — Object table number. This parameter is only used when the *ObjType* is [SR Object Type Field](#).

FieldNo — Object field number. This parameter is only used when the *ObjType* is [SR Object Type Field](#).

VarType — Object variable type. This parameter is only used when the *ObjType* is [SR Object Type Variable](#). You may choose one of the following variable types

- 1 - [SR Variable Type Variable](#)
- 2 - [SR Variable Type Array Auto](#)
- 3 - [SR Variable Type Array Element](#)

ArrayElem — Desired array element when the *ObjType* is [SR Object Type Variable](#) and the *VarType* is [SR Variable Type Array Element](#).

CalcType — Desired calculation option. This parameter is only used when the *ObjType* is SR Object Type Field or SR Object Type Variable. You may choose one of the following values

- 0 - SR Calculation Type None
- 1 - SR Calculation Type Total
- 2 - SR Calculation Type Min
- 3 - SR Calculation Type Average
- 4 - SR Calculation Type Max

CalcName — Name of the variable to store calculation. This parameter is only used when the *Options* is SR Obj Flag Record Calc Value.

Rows — Number of rows in a rectangle grid. This parameter is only used when the *ObjType* is SR Object Type Rectangle.

Cols — Number of columns in a rectangle grid. This parameter is only used when the *ObjType* is SR Object Type Rectangle.

Repeat Horizontal Offset — Desired horizontal offset for repeating object. This parameter is only used when the *Options* is SR Obj Flag Repeat Horizontally

Repeat Vertical Offset — Desired vertical offset for repeating object. This parameter is only used when the *Options* is SR Obj Flag Repeat Vertically

-> *Result Code* —If the routine completed successfully, a valid ItemID (greater than zero) will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will create a new report (see SR New Offscreen Area), then create two new fields and finally save the report to disk as "test.srp".

```
C_LONGINT($SR_AREA)
C_LONGINT($SR_Err;$SR_ItemID)

$SR_Err:=0
$SR_ItemID:=0

$SR_AREA:=SR New Offscreen Area
If ($SR_AREA>0)

    `add field
    $SR_ObjName:=""
    $SR_ObjType:=SR Object Type Field
    $SR_Opts:=0
    $SR_Selected:=0
    $SR_VarType:=0
    $SR_ArrayElement:=0
    $SR_CalcType:=0
    $SR_CalcName:=""
    $SR_Rows:=0
    $SR_Cols:=0
    $SR_HRepeat:=0
    $SR_VRepeat:=0

    $offset:=5
    $top:=40
    $left:=40
    $bottom:=$top+12
    $right:=$left+50

    $SR_ItemID:=SR Create Object($SR_AREA;$SR_ObjName;$top;$left;$bottom;$right;$SR_ObjType;
        $SR_Opts;$SR_Selected;Table(->[Customer]Address);Field(>[Customer]Address);
        $SR_VarType;$SR_ArrayElement;$SR_CalcType;$SR_CalcName;$SR_Rows;$SR_Cols;
        $SR_HRepeat;$SR_VRepeat)

    $left:=$right+$offset
    $right:=$left+50
    $SR_ItemID:=SR Create Object ($SR_AREA;$SR_ObjName;$top;$left;$bottom;$right;$SR_ObjType;
        $SR_Opts;$SR_Selected;Table(->[Customer]Name);Field(->[Customer]Name);
        $SR_VarType;$SR_ArrayElement;$SR_CalcType;$SR_CalcName;$SR_Rows;$SR_Cols;
        $SR_HRepeat;$SR_VRepeat)

    $SR_Err:=SR Save Report ($SR_AREA;"test.srp";0)
    SR DELETE OFFSCREEN AREA ($SR_AREA)

End if
```

See Also: SR Set Object Data, SR Object Format, SR Delete Object.

SR Delete Object

SR Delete Object (reportArea:L; objectID:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
ObjectID	C_LONGINT	Report Object ID
-> ResultCode	C_LONGINT	Result Code

SR Delete Object provides the ability to procedurally delete an object from a report. After the object is deleted, the internal object list is updated automatically.

ReportArea — The desired SuperReport Pro area.

Object ID — Desired ObjectID you wish to delete. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Properties**, etc.) to obtain detailed information about each report object.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example deletes a SuperReport Pro object.

```
$SR_Err:=SR Delete Object($SR_AREA;$SR_ObjectID)
```

See Also: You may delete any object which you have created procedurally SR Create Object or objects which have been created by users in the report editor.

SR Get Object Scripts

SR Get Object Scripts (reportArea:L; objectID:L; sectionScript:T; htmlPre:T; htmlPost:T) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Object ID	C_LONGINT	Report section ID
Object Script	C_TEXT	Object Script
Object HTML Start	C_TEXT	HTML Start Block
Object HTML End	C_TEXT	HTML End Block
-> ResultCode	C_LONGINT	Result Code

SR Get Object Scripts will retrieve the object script and HTML information for the defined object.

ReportArea — The desired SuperReport Pro area.

Object ID — Desired ObjectID. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Properties**, etc.) to obtain detailed information about each report object.

Object Script — A valid 4th Dimension variable, which will receive the object script.

Object HTML Start — A valid 4th Dimension variable, which will receive the objects start HTML block.

Object HTML End — A valid 4th Dimension variable, which will receive the objects end HTML block.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will retrieve the object script and HTML data.

C_TEXT(tSR_Script;tSR_HTMLStart;tSR_HTMLEnd)

\$SR_Err:=SR Get Object Scripts(\$SR_AREA;\$SR_ObjectID;tSR_Script;tSR_HTMLStart;tSR_HTMLEnd)

See Also: SR Set Object Scripts, SR Get Object Data, SR Get Object Format, SR Get Object Properties.

SR Set Object Scripts

SR Set Object Scripts (reportArea:L; objectID:L; objectScript:T; htmlPre:T; htmlPost:T) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Object ID	C_LONGINT	Report section ID
Object Script	C_TEXT	Object Script
Object HTML Start	C_TEXT	HTML Start Block
Object HTML End	C_TEXT	HTML End Block
-> ResultCode	C_LONGINT	Result Code

SR Set Object Scripts will set the object script and HTML information for the defined object.

ReportArea — The desired SuperReport Pro area.

Object ID — Desired ObjectID you wish to set the object script and HTML data. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Properties**, etc.) to obtain detailed information about each report object.

Object Script — Desired object script.

Object HTML Start — Desired object HTML start block.

Object HTML End — Desired object HTML end block.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will set the object script and HTML data.

C_TEXT(tSR_Script;tSR_HTMLStart;tSR_HTMLEnd)

```
tSR_Script:="sPhone:=FormatPhone([Customer]Phone)
tSR_HTMLStart:="<td>"
tSR_HTMLEnd:="</td>"
```

```
$SR_Err:=SR Set Object Scripts($SR_AREA;$SR_ObjectID ;tSR_Script;tSR_HTMLStart;tSR_HTMLEnd)
```

See Also: SR Get Object Scripts, SR Set Object Data, SR Set Object Format, SR Set Object Properties.

SR Get Object Properties

SR Get Object Properties (reportArea:L; objectID:L; objName:S; rectTop:L; rectLeft:L; rectBottom:L; rectRight:L; objType:L; options:L; order:L; selected:L; tableNo:L; fieldNo:L; varType:L; arrayElem:L; calcType:L; calcName:S; rows:L; cols:L; repeatHOffset:L; repeatVOffset:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Object ID	C_LONGINT	Report section ID
ObjName	C_STRING or C_TEXT	Object Name
RectTop	C_LONGINT	Object Position – Top
RectLeft	C_LONGINT	Object Position – Left
RectBottom	C_LONGINT	Object Position – Bottom
RectRight	C_LONGINT	Object Position – Right
ObjType	C_LONGINT	Object Type
Options	C_LONGINT	Object Options
Order	C_LONGINT	Object Order
Selected	C_LONGINT	Object Selected
TableNo	C_LONGINT	Object Table Number
FieldNo	C_LONGINT	Object Field Number
VarType	C_LONGINT	Object Variable Type
ArrayElem	C_LONGINT	Array Element
CalcType	C_LONGINT	Object Calc Type
CalcName	C_STRING or C_TEXT	Object Calc Name
Rows	C_LONGINT	Object Rows
Cols	C_LONGINT	Object Cols
RepeatHOffset	C_LONGINT	Object Horizontal Repeat Offset
RepeatVOffset	C_LONGINT	Object Vertical Repeat Offset
-> ResultCode	C_LONGINT	Result Code

SR Get Object Properties provides the ability to procedurally retrieve object properties. When used with **SR Get Object Format**, **SR Get Object Data** and **SR Get Object Scripts**, you can obtain all information about a given object.

Note: All parameters must be valid 4th Dimension variables (you cannot use 4D local variables). You may not receive information into 4th Dimension fields.

ReportArea — The desired SuperReport Pro area.

ObjectID — Desired objectID you wish to obtain object properties. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Format**, etc.) to obtain detailed information about each report object.

Object Name — Contains the field alias (see SR Structure) or the Variable name (if you have created a variable). If you wish to get static text information, please use **SR Get Object Data**.

RectTop — Object Position – Top. The object position is relative to the upper left corner of the report area.

RectLeft — Object Position – Left. The object position is relative to the upper left corner of the report area.

RectBottom — Object Position – Bottom. The object position is relative to the upper left corner of the report area.

RectRight — Object Position – Right. The object position is relative to the upper left corner of the report area.

Object Type — Desired object type. One of the following values will be returned.

1 - SR Object Type Text

- 2 - [SR Object Type Field](#)
- 3 - [SR Object Type Variable](#)
- 4 - [SR Object Type Line](#)
- 5 - [SR Object Type Rectangle](#)
- 6 - [SR Object Type Circle](#)
- 8 - [SR Object Type Picture](#)

If you are retrieving properties for a static text object ([SR Object Type Text](#)) or picture object ([SR Object Type Picture](#)) you can retrieve the static information using **SR Get Object Data** after this routine has completed.

Options — Object Options. The value returned by this parameter will be the sum of the object options. You can use 4Ds standard bitwise operators to extract individual item options.

- 1 - [SR Obj Flag Left Line](#)
- 2 - [SR Obj Flag Top Line](#)
- 4 - [SR Obj Flag Right Line](#)
- 8 - [SR Obj Flag Bottom Line](#)
- 15 - [SR Obj Flag All Lines](#)
- 16 - [SR Obj Flag Fixed Horizontal](#)
- 32 - [SR Obj Flag Fixed Vertical](#)
- 64 - [SR Obj Flag Grow Horizontal](#)
- 128 - [SR Obj Flag Grow Vertical](#)
- 256 - [SR Obj Flag Variable Width](#)
- 512 - [SR Obj Flag Variable Height](#)
- 1024 - [SR Obj Flag Replace If Empty](#)
- 2048 - [SR Obj Flag Use Calc Value](#)
- 4096 - [SR Obj Flag Show Calc Value](#)
- 8192 - [SR Obj Flag Repeating Object](#)
- 16384 - [SR Obj Flag Repeat Vertically](#)
- 32768 - [SR Obj Flag Repeat Horizontally](#)
- 65536 - [SR Obj Flag Replace Row If Empty](#)

Order — This is the value users see when “Show Object Order” is selected when using the report editor. It can be used to determine current object order (starting at one).

Selected — Determines if the object is current selected within the report area.

- 0 – Object is not selected
- 1 – Object is selected

TableNo — Object table number. This parameter is only used when the *ObjType* is [SR Object Type Field](#).

FieldNo — Object field number. This parameter is only used when the *ObjType* is [SR Object Type Field](#).

VarType — Object variable type. This parameter is only used when the *ObjType* is SR Object Type Variable. You may choose one of the following variable types

- 1 - SR Variable Type Variable
- 2 - SR Variable Type Array Auto
- 3 - SR Variable Type Array Element

ArrayElem — Desired array element when the *ObjType* is SR Object Type Variable and the *VarType* is SR Variable Type Array Element.

CalcType — Desired calculation option. This parameter is only used when the *ObjType* is SR Object Type Field or SR Object Type Variable. You may choose one of the following values

- 0 - SR Calculation Type None
- 1 - SR Calculation Type Total
- 2 - SR Calculation Type Min
- 3 - SR Calculation Type Average
- 4 - SR Calculation Type Max

CalcName — Name of the variable to store calculation. This parameter is only used when the *Options* is SR Obj Flag Use Calc Value.

Rows — Number of rows in a rectangle grid. This parameter is only used when the *ObjType* is SR Object Type Rectangle.

Cols — Number of columns in a rectangle grid. This parameter is only used when the *ObjType* is SR Object Type Rectangle.

Repeat Horizontal Offset — Desired horizontal offset for repeating object. This parameter is only used when the *Options* is SR Obj Flag Repeat Horizontally

Repeat Vertical Offset — Desired vertical offset for repeating object. This parameter is only used when the *Options* is SR Obj Flag Repeat Vertically

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

See Also: SR Get Object Scripts, SR Get Object Data, SR Get Object Format.

SR Set Object Properties

SR Set Object Properties (reportArea:L; objectID:L; action:L; objName:S; rectTop:L; rectLeft:L; rectBottom:L; rectRight:L; objType:L; options:L; order:L; selected:L; tableNo:L; fieldNo:L; varType:L; arrayElem:L; calcType:L; calcName:S; rows:L; cols:L; repeatHOffset:L; repeatVOffset:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Object ID	C_LONGINT	Report section ID
Action	C_LONGINT	Action
ObjName	C_STRING or C_TEXT	Object Name
RectTop	C_LONGINT	Object Position – Top
RectLeft	C_LONGINT	Object Position – Left
RectBottom	C_LONGINT	Object Position – Bottom
RectRight	C_LONGINT	Object Position – Right
ObjType	C_LONGINT	Object Type
Options	C_LONGINT	Object Options
Order	C_LONGINT	Object Order
Selected	C_LONGINT	Object Selected
TableNo	C_LONGINT	Object Table Number
FieldNo	C_LONGINT	Object Field Number
VarType	C_LONGINT	Object Variable Type
ArrayElem	C_LONGINT	Array Element
CalcType	C_LONGINT	Object Calc Type
CalcName	C_STRING or C_TEXT	Object Calc Name
Rows	C_LONGINT	Object Rows
Cols	C_LONGINT	Object Cols
RepeatHOffset	C_LONGINT	Object Horizontal Repeat Offset
RepeatVOffset	C_LONGINT	Object Vertical Repeat Offset
-> ResultCode	C_LONGINT	Result Code

SR Set Object Properties provides the ability to procedurally set individual object properties. When used with **SR Set Object Format**, **SR Set Object Data** and **SR Set Object Scripts**, you can set all information about a given object.

If you are setting the properties for a static text object ([SR Object Type Text](#)) or picture object ([SR Object Type Picture](#)) you can set the static information using **SR Set Object Data** after this routine has completed.

ReportArea — The desired SuperReport Pro area.

ObjectID — Desired objectID you wish to obtain object properties. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Properties**, etc.) to obtain detailed information about each report object.

Action — Desired property you wish to set. If you are setting multiple attributes, you should pass the accumulative total of all action codes.

You may choose one or more of the following selectors. If you are setting all attributes, you can use the SR Property All selector.

Property Name	Description
-1 - <u>SR Property All</u>	Set all object properties
1 - <u>SR Property Name</u>	Set the objName (set variable/field)
2 - <u>SR Property Position</u>	Set rectTop, rectLeft, rectBottom, rectRight
4 - <u>SR Property Type</u>	Set type
8 - <u>SR Property Options</u>	Set options
16 - <u>SR Property Selected</u>	Set selected
32 - <u>SR Property Field</u>	Set table & field, set name if is <u>SR Property Name</u> also set (field objects only)
64 - <u>SR Property Variable Type</u>	Set varType (variable objects only)
128 - <u>SR Property Calculation</u>	Set calcType & calcName (variable/field objects only)
256 - <u>SR Property Rows Cols</u>	Set rows & cols for (rect objects only)
512 - <u>SR Property Repeat Offsets</u>	Set repeatHOffset & repeatVOffset

Object Name — Desired object name. If you are referencing an object, which provides static information (i.e. a static text object or variable), parameter will contain the static information.

RectTop — Object Position – Top. The object position is relative to the upper left corner of the report area.

RectLeft — Object Position – Left. The object position is relative to the upper left corner of the report area.

RectBottom — Object Position – Bottom. The object position is relative to the upper left corner of the report area

RectRight — Object Position – Right. The object position is relative to the upper left corner of the report area.

Object Type — Desired object type. One of the following values will be returned.

- 1 - SR Object Type Text
- 2 - SR Object Type Field
- 3 - SR Object Type Variable
- 4 - SR Object Type Line
- 5 - SR Object Type Rectangle
- 6 - SR Object Type Circle
- 8 - SR Object Type Picture

Options — Object Options. The value returned by this parameter will be the sum of the object options. You can use 4Ds standard bitwise operators to extract individual item options.

- 1 - [SR Obj Flag Left Line](#)
- 2 - [SR Obj Flag Top Line](#)
- 4 - [SR Obj Flag Right Line](#)
- 8 - [SR Obj Flag Bottom Line](#)
- 15 - [SR Obj Flag All Lines](#)
- 16 - [SR Obj Flag Fixed Horizontal](#)
- 32 - [SR Obj Flag Fixed Vertical](#)
- 64 - [SR Obj Flag Grow Horizontal](#)
- 128 - [SR Obj Flag Grow Vertical](#)
- 256 - [SR Obj Flag Variable Width](#)
- 512 - [SR Obj Flag Variable Height](#)
- 1024 - [SR Obj Flag Replace If Empty](#)
- 2048 - [SR Obj Flag Use Calc Value](#)
- 4096 - [SR Obj Flag Show Calc Value](#)
- 8192 - [SR Obj Flag Repeating Object](#)
- 16384 - [SR Obj Flag Repeat Vertically](#)
- 32768 - [SR Obj Flag Repeat Horizontally](#)
- 65536 - [SR Obj Flag Replace Row If Empty](#)

Order — Object order (position in object list). This value is ignored when using SR Set Object Properties. If you wish to reorder an object, use **SR Change Object Order** or **SR Reorder Objects**.

Selected — Determines if the object is current selected within the report area.

- 0 – Object is not selected
- 1 – Object is selected

TableNo — Object table number. This parameter is only used when the *ObjType* is [SR Object Type Field](#).

FieldNo — Object field number. This parameter is only used when the *ObjType* is [SR Object Type Field](#).

VarType — Object variable type. This parameter is only used when the *ObjType* is [SR Object Type Variable](#). You may choose one of the following variable types

- 1 - [SR Variable Type Variable](#)
- 2 - [SR Variable Type Array Auto](#)
- 3 - [SR Variable Type Array Element](#)

ArrayElem — Desired array element when the *ObjType* is [SR Object Type Variable](#) and the *VarType* is [SR Variable Type Array Element](#).

CalcType — Desired calculation option. This parameter is only used when the *ObjType* is SR Object Type Field or SR Object Type Variable. You may choose one of the following values

- 0 - SR Calculation Type None
- 1 - SR Calculation Type Total
- 2 - SR Calculation Type Min
- 3 - SR Calculation Type Average
- 4 - SR Calculation Type Max

CalcName — Name of the variable to store calculation. This parameter is only used when the *Options* is SR Obj Flag Record Calc Value.

Rows — Number of rows in a rectangle grid. This parameter is only used when the *ObjType* is SR Object Type Rectangle.

Cols — Number of columns in a rectangle grid. This parameter is only used when the *ObjType* is SR Object Type Rectangle.

Repeat Horizontal Offset — Desired horizontal offset for repeating object. This parameter is only used when the *Options* is SR Obj Flag Repeat Horizontally

Repeat Vertical Offset — Desired vertical offset for repeating object. This parameter is only used when the *Options* is SR Obj Flag Repeat Vertically

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

See Also: SR Set Object Scripts, SR Set Object Data, SR Set Object Format.

SR Get Object Format

SR Get Object Format (reportArea:L; objectID:L; fontName:S; fontSize:L; fontStyle:L; justification:L; formatString:S; foreRed:L; foreGreen:L; foreBlue:L; backRed:L; backGreen:L; backBlue:L; forePattern:L; backPattern:L; lineThickness:L; foreColor:L; backColor:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Object ID	C_LONGINT	Report section ID
FontName	C_STRING or C_TEXT	Font Name
FontSize	C_LONGINT	Font Size
FontStyle	C_LONGINT	Font Style
Justification	C_LONGINT	Justification
Format String	C_LONGINT	Object Formatting String
ForeRed	C_LONGINT	Foreground – Red
ForeGreen	C_LONGINT	Foreground – Green
ForeBlue	C_LONGINT	Foreground – Blue
BackRed	C_LONGINT	Background – Red
BackGreen	C_LONGINT	Background – Green
BackBlue	C_LONGINT	Background – Blue
ForePattern	C_LONGINT	Foreground Pattern
BackPattern	C_LONGINT	Background Pattern
Line Thickness	C_LONGINT	Line Thickness
ForeColor	C_LONGINT	Fore Color*
BackColor	C_LONGINT	Back Color*

* ForeColor and BackColor are 4D's style of expressing RGB colors (0x00RRGGBB)

-> ResultCode C_LONGINT Result Code

SR Get Object Format provides the ability to procedurally retrieve object-formatting properties. When used with **SR Get Object Properties**, **SR Get Object Data** and **SR Get Object Scripts**, you can obtain all information about a given object.

Note: All parameters must be valid 4th Dimension variables (you cannot use 4D fields).

ReportArea — The desired SuperReport Pro area.

ObjectID — Desired objectID you wish to obtain object properties. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Properties**, etc.) to obtain detailed information about each report object. In addition, passing a value of –1 will retrieve the “default” report attributes.

Font Name — Desired object font name.

Font Size — Desired objects font size (1 – 144).

Font Style — Object font style. Uses Macintosh style of style attributes. Use 4D constants (see FONT STYLE).

Font Justification — Object font justification. You may use one of the following values

- 1 – Left
- 2 – Center
- 3 – Right

Format String — Object format string. This format string is used when printing as the display format.

ForeRed — Object foreground color (Red). You can also use the *ForeColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

ForeGreen — Object foreground color (Green). You can also use the *ForeColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

ForeBlue — Object foreground color (Blue). You can also use the *ForeColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

BackRed — Object background color (Red). You can also use the *BackColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

BackGreen — Object background color (Green). You can also use the *BackColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

BackBlue — Object background color (Blue). You can also use the *BackColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

ForePattern — Object foreground pattern (0-32).

BackPattern — Object background pattern (0-32).

Line Thickness — Object line thickness (when SR Object Type Line, SR Object Type Circle, or SR Object Type Rectangle). Valid range of 1-9 (1 equals hairline).

ForeColor — Object foreground color using 4Ds style of expressing RGB colors (0x00RRGGBB)

BackColor — Object background color using 4Ds style of expressing RGB colors (0x00RRGGBB)

See Also: SR Get Object Scripts, SR Get Object Data, SR Set Object Format, SR Get Object Properties.

SR Set Object Format

SR Set Object Format (reportArea:L; objectID:L; action:L; fontName:S; fontSize:L; fontStyle:L; justification:L; formatString:S; foreRed:L; foreGreen:L; foreBlue:L; backRed:L; backGreen:L; backBlue:L; forePattern:L; backPattern:L; lineThickness:L; foreColor:L; backColor:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Object ID	C_LONGINT	Report section ID
Action	C_LONGINT	Action
FontName	C_STRING or C_TEXT	Font Name
FontSize	C_LONGINT	Font Size
FontStyle	C_LONGINT	Font Style
Justification	C_LONGINT	Justification
Format String	C_LONGINT	Object Formatting String
ForeRed	C_LONGINT	Foreground – Red
ForeGreen	C_LONGINT	Foreground – Green
ForeBlue	C_LONGINT	Foreground – Blue
BackRed	C_LONGINT	Background – Red
BackGreen	C_LONGINT	Background – Green
BackBlue	C_LONGINT	Background – Blue
ForePattern	C_LONGINT	Foreground Pattern
BackPattern	C_LONGINT	Background Pattern
Line Thickness	C_LONGINT	Line Thickness
ForeColor	C_LONGINT	Fore Color*
BackColor	C_LONGINT	Back Color*

* ForeColor and BackColor are 4D's style of expressing RGB colors (0x00RRGGBB)

-> ResultCode C_LONGINT Result Code

SR Set Object Format provides the ability to procedurally set object-formatting properties. When used with **SR Set Object Properties**, **SR Set Object Data** and **SR Set Object Scripts**, you can obtain all information about a given object.

In addition to using this routine in you own object interface, you may also use this routine during printing to procedurally override the default appear of objects while printing. For example, if you wanted to set the appearance of a given object based on the current value, you could use SR Set Object Format to perform this action.

ReportArea — The desired SuperReport Pro area.

ObjectID — Desired objectID you wish to obtain object properties. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Properties**, etc.) to obtain detailed information about each report object. In addition, passing a value of -1 will retrieve the “default” report attributes.

Action — Desired property you wish to set. If you are setting multiple attributes, you should pass the accumulative total of all action codes.

You may choose one or more of the following selectors. If you are setting all attributes, you can use the [SR Property All](#) selector.

Property Name	Description
-1 - SR Property All	Set all object properties
1 - SR Attribute Font Name	Set the font name
2 - SR Attribute Fore Color	Set the foreground color using ForeRed, ForeGreen, ForeBlue
4 - SR Attribute Back Color	Set the background color using BackRed, BackGreen, BackBlue
8 - SR Attribute Font Size	Set the font size
16 - SR Attribute Font Style	Set the font style
32 - SR Attribute Fore Pattern	Set the foreground pattern
64 - SR Attribute Back Pattern	Set the background pattern
128 - SR Attribute Justification	Set the justification
256 - SR Attribute Line Thickness	Set the line thickness
512 - SR Attribute Format	Set the object format string
1024 - SR Attribute 4D Fore Color	Set foreground color using ForeColor
2048 - SR Attribute 4D Back Color	Set background color using BackColor
4096 - SR Attribute No Adjust	Don't adjust bounding boxes (for fields, variables, etc.)

Font Name — Desired object font name.

Font Size — Desired objects font size (1 – 144).

Font Style — Object font style. Uses Macintosh style of style attributes. Use 4D constants (see FONT STYLE).

Font Justification — Object font justification. You may use one of the following values

- 1 – Left
- 2 – Center
- 3 – Right

Format String — Object format string. This format string is used when printing as the display format. The format string can only be set on valid objects (ObjectID greater than zero) and will be ignored if you are setting the default attributes (ObjectID –1) and a result value of –8007 will be returned.

ForeRed — Object foreground color (Red). You can also use the *ForeColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

ForeGreen — Object foreground color (Green). You can also use the *ForeColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

ForeBlue — Object foreground color (Blue). You can also use the *ForeColor* parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

BackRed — Object background color (Red). You can also use the BackColor parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

BackGreen — Object background color (Green). You can also use the BackColor parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

BackBlue — Object background color (Blue). You can also use the BackColor parameter, which returns the foreground color in 4Ds style of expressing RGB colors.

ForeColor — Object foreground color (0-32).

BackColor — Object background color (0-32).

Line Thickness — Object line thickness (when SR Object Type Line or SR Object Type Rectangle). Valid range of 1-9 (1 equals hairline).

ForeColor — Object foreground color using 4Ds style of expressing RGB colors (0x00RRGGBB)

BackColor — Object background color using 4Ds style of expressing RGB colors (0x00RRGGBB)

SR Get Object Data

SR Get Object Data (reportArea:L; objectID:L; staticPICT:P; staticText:T) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Object ID	C_LONGINT	Report section ID
StaticPICT	C_PICTURE	Object Static PICT
StaticText	C_TEXT	Object Static TEXT
-> ResultCode	C_LONGINT	Result Code

SR Get Object Data will retrieve the static object information (SR Object Type PICT or SR Object Type Text). When used with **SR Get Object Properties**, **SR Get Object Format** and **SR Set Object Scripts**, you can obtain all information about a given object.

ReportArea — The desired SuperReport Pro area.

Object ID — Desired ObjectID. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Properties**, etc.) to obtain detailed information about each report object.

Static PICT— A valid 4th Dimension variable, which will receive the static PICT data for a SR Object Type PICT.

Static Text— A valid 4th Dimension variable, which will receive the static TEXT data for a SR Object Type Text. In addition, if you have assigned HTML text to a PICT object, the HTML information will be returned in this parameter as well.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will retrieve the static text or pict information.

C_TEXT(tSR_TextData)

C_PICT(pSR_PICTData)

\$SR_Err:=**SR Get Object Data**(\$SR_AREA;\$SR_ObjectID;pSR_PICTData;tSR_TextData)

See Also: SR Get Object Scripts, SR Set Object Data, SR Get Object Format, SR Get Object Properties.

SR Set Object Data

SR Set Object Data (reportArea:L; objectID:L; staticPICT:P; staticText:T) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Object ID	C_LONGINT	Report section ID
StaticPICT	C_PICTURE	Object Static PICT
StaticText	C_TEXT	Object Static TEXT
-> ResultCode	C_LONGINT	Result Code

SR Set Object Data will set the static object information (SR Object Type PICT or SR Object Type Text). When used with **SR Set Object Properties**, **SR Set Object Format** and **SR Set Object Scripts**, you can obtain all information about a given object.

ReportArea — The desired SuperReport Pro area.

Object ID — Desired ObjectID. You can use the object support routines (**SR Get Object IDs**, **SR Get Object Properties**, etc.) to obtain detailed information about each report object.

Static PICT— A valid 4th Dimension variable containing the PICT information you wish to assign to a static PICT object (SR Object Type PICT).

Static Text— Desired text you wish to assign to a static text object (SR Object Type Text) or HTML data you wish to assign to a PICT object (SR Object Type PICT).

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

See Also: SR Set Object Scripts, SR Get Object Data, SR Set Object Format, SR Set Object Properties.

SR Get Guides

SR Get Guides (reportArea:L; horCoordinates:X; vertCoordinates:X) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Horz Coordinates	ARRAY INTEGER or LONGINT	Array of horizontal coordinates
Vert Coordinates	ARRAY INTEGER or LONGINT	Array of vertical coordinates
-> ResultCode	C_LONGINT	Result Code

SR Get Guides builds a 4th Dimension array containing the horizontal and vertical coordinates for all report guides.

ReportArea — The desired SuperReport Pro area.

Horizontal Coordinates — A valid 4th Dimension array which will receive a list of vertical positions for all horizontal guides.

Vertical Coordinates — A valid 4th Dimension array which will receive a list of horizontal positions for all vertical guides.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will build arrays of horizontal and vertical coordinates for the report guides.

```
ARRAY INTEGER(aiSR_HGuides;0)
ARRAY INTEGER(aiSR_VGuides;0)
$SR_Err:=SR Get Guides($SR_AREA;aiSR_HGuides;aiSR_VGuides)
```

See Also: SR Set Guides.

SR Set Guides

SR Set Guides (reportArea:L; horCoordinates:X; vertCoordinates:X) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Horz Coordinates	ARRAY INTEGER or LONGINT	Array of horizontal coordinates
Vert Coordinates	ARRAY INTEGER or LONGINT	Array of vertical coordinates
-> ResultCode	C_LONGINT	Result Code

SR Set Guides will create report guides using the supplied horizontal and vertical coordinate arrays.

ReportArea — The desired SuperReport Pro area.

Horizontal Coordinates — A valid 4th Dimension array, which contains the horizontal, coordinates for the report guides you wish to create.

Vertical Coordinates — A valid 4th Dimension array, which contains the vertical, coordinates for the report guides you wish to create.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will create two report guides using the horizontal and vertical coordinates.

```
ARRAY INTEGER($aiSR_HGuides;1)  
ARRAY INTEGER($aiSR_VGuides;3)
```

```
aiSR_HGuides{1}:=100  
aiSR_VGuides{1}:=50  
aiSR_VGuides{2}:=150  
aiSR_VGuides{3}:=250
```

```
$SR_Err:=SR Set Guides($SR_AREA;$aiSR_HGuides;$aiSR_VGuides)
```

See **Also**: SR Get Guides.

SR Get Scripts

SR Get Scripts (reportArea:L; startScript:T; bodyScript:T; endScript:T) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Start Script	C_TEXT	Report Start Script
Body Script	C_TEXT	Report Body Script
End Script	C_TEXT	Report End Script
-> ResultCode	C_LONGINT	Result Code

SR Get Scripts will return each of the scripts associated to the report area (configured in database menu).

ReportArea — The desired SuperReport Pro area.

Start Script — A valid 4th Dimension variable, which will receive the script, associated to the **Start** script.

Body Script — A valid 4th Dimension variable, which will receive the script, associated to the **Body** script.

End Script — A valid 4th Dimension variable, which will receive the script, associated to the End script.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will return the database scripts (Start, Body, and End) for the desired report area.

C_TEXT(tSR_StartScript;tSR_BodyScript;tSR_EndScript)

\$SR_Err:=SR Get Scripts(\$SR_AREA;tSR_StartScript;tSR_BodyScript;tSR_EndScript)

See Also: SR Set Scripts.

SR Set Scripts

SR Set Scripts (reportArea:L; startScript:T; bodyScript:T; endScript:T) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Start Script	C_TEXT	Report Start Script
Body Script	C_TEXT	Report Body Script
End Script	C_TEXT	Report End Script
-> ResultCode	C_LONGINT	Result Code

SR Set Scripts will set each of the scripts associated to the report area (configured in database menu).

ReportArea — The desired SuperReport Pro area.

Start Script — The desired text, which contains the **Start** script.

Body Script — The desired text, which contains the **Body** script.

End Script — The desired text, which contains the **End** script.

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will return the database scripts (Start, Body, and End) for the desired report area.

C_TEXT(tSR_StartScript;tSR_BodyScript;tSR_EndScript)

```
tSR_StartScript:="ALL RECORDS([Customer])"  
tSR_BodyScript:="" `no body script  
tSR_EndScript:="UNLOAD RECORD([Customer])"
```

```
$SR_Err:=SR Set Scripts($SR_AREA;tSR_StartScript;tSR_BodyScript;tSR_EndScript)
```

See Also: SR Get Scripts.

SR Get Sections

SR Get Sections (reportArea:L; activeSections:L; printSections:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Active Sections	C_LONGINT	Active Sections Value
Print Sections	C_LONGINT	Print Sections Value
-> ResultCode	C_LONGINT	Result Code

SR Get Sections will return the active and print sections, which have been set for the desired report area. The value returned to the section variables is a bitwise representation of the accumulative sections.

You can use the SuperReport Pro Section Constants to determine if a selected section is active by using the SR Section Mask attributes:

- [SR Section Header Mask](#)
- [SR Section SubHeader1 Mask](#)
- [SR Section SubHeader2 Mask](#)
- [SR Section SubHeader3 Mask](#)
- [SR Section SubHeader4 Mask](#)
- [SR Section SubHeader5 Mask](#)
- [SR Section SubHeader6 Mask](#)
- [SR Section Body Mask](#)
- [SR Section SubTotal6 Mask](#)
- [SR Section SubTotal5 Mask](#)
- [SR Section SubTotal4 Mask](#)
- [SR Section SubTotal3 Mask](#)
- [SR Section SubTotal2 Mask](#)
- [SR Section SubTotal1 Mask](#)
- [SR Section Total Mask](#)
- [SR Section Footer Mask](#)

For complete details on valid section codes, please refer to **Chapter 7 — SuperReport Pro Codes (Section Codes)**

ReportArea — The desired SuperReport Pro area.

Active Sections — A valid 4th Dimension variable which will receive the bitwise value of current active sections.

Print Sections — A valid 4th Dimension variable which will receive the bitwise value of current print sections.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will retrieve the current active and print sections for the desired report area.

```
C_LONGINT(iSR_Sections;iSR_PrintSections)
```

```
iSR_Sections:=0
```

```
iSR_PrintSections:=0
```

```
$SR_Err:=SR Get Sections($SR_AREA;iSR_Sections;iSR_PrintSections)
```

Using the SR Section mask values, you can easily determine if a given section has been activated using the following syntax:

```
If (iSR_Sections & SR Section SubTotal1 Mask)
```

```
  `SubHeader1 is active
```

```
End if
```

```
If (iSR_PrintSections & SR Section Body Mask)
```

```
  `Body section will be used when printing
```

```
End if
```

If the supplied section was active, a value of one will be returned; otherwise a value of zero will be returned.

See Also: SR Set Sections, SR Get Section Properties.

SR Set Sections

SR Set Sections (reportArea:L; activeSections:L; printSections:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Active Sections	C_LONGINT	Active Sections Value
Print Sections	C_LONGINT	Print Sections Value
-> ResultCode	C_LONGINT	Result Code

SR Set Sections will set the active and print sections for the desired report area. The values passed to each section parameter are bitwise representation of the accumulative sections.

When activating sections, the Header, Body, Footer and Total sections will always be active, regardless of any settings you make to the contrary as they are required sections which can't be removed.

For complete details on valid section codes, please refer to **Chapter 7 — SuperReport Pro Codes (Section Codes)**

ReportArea — The desired SuperReport Pro area.

Active Sections — The desired sections you would like available on the report area.

Print Sections — The desired sections you would like available on the report area.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will enable Subheader 1 and SubTotal 1.

C_LONGINT(iSR_Sections;iSR_PrintSections)

iSR_Sections:= SR Section SubHeader1 Mask + SR Section SubTotal1 Mask

iSR_PrintSections:= 0 `use all available print sections

\$SR_Err:=**SR Set Sections**(\$SR_AREA;iSR_Sections;iSR_PrintSections)

See Also: SR Get Sections, SR Set Section Properties.

SR Get Section IDs

SR Get Section IDs (reportArea:L; sectionID:L; sectionID:X) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Section ID	C_LONGINT	Report Section ID to obtain properties
Section ID List	ARRAY LONGINT	Array of section IDs
-> ResultCode	C_LONGINT	Result Code

SR Get Section IDs will retrieve all the information about a given section object. While not all properties are related to one another, this single routine enables you to easily retrieve all information for a given section.

ReportArea — The desired SuperReport Pro area.

Section ID — A valid SuperReport Pro section (i.e. SR Section Body or SR Section Header). For a complete list of sections, please refer to **Chapter 7 — SuperReport Pro Codes (Section Codes)**.

Section ID List — A valid 4th Dimension array which will contain the Object ID's for all the objects in the defined section.

Example

The following example will return an array of all the objects in the body section.

ARRAY LONGINT(aiSR_SectionID;0)

iSR_Err:=**SR Get Section IDs** (vSRArea;SR Section Body;aiSR_SectionID)

See Also: SR Get Sections, SR Set Section Properties, SR Set Section Scripts.

SR Get Section Properties

SR Get Section Properties (reportArea:L; sectionID:L; useSection:L; print:L; position:L; options:L; throwPage:L; minSpace:L; breakType:L; breakTable:L; breakField:L; breakVarName:S) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Section ID	C_LONGINT	Report Section ID to obtain properties
Use Section	C_LONGINT	Use section flag
Print Section	C_LONGINT	Print section flag
Position	C_LONGINT	Section vertical position
Options	C_LONGINT	Miscellaneous Options
Throw Page	C_LONGINT	Start New Page Setting
Min Space	C_LONGINT	Minimum Space
Break Type	C_LONGINT	Section break type
Break Table	C_LONGINT	Break Table Number (when break on field)
Break Field	C_LONGINT	Break Field Number (when break on field)
Break VarName	C_STRING or C_TEXT	Break Variable Name (when break on var)
-> ResultCode	C_LONGINT	Result Code

SR Get Section Properties will retrieve all the information about a given section object. While not all properties are related to one another, this single routine enables you to easily retrieve all information for a given section.

ReportArea — The desired SuperReport Pro area.

Section ID — A valid SuperReport Pro section (i.e. SR Section Body or SR Section Header). For a complete list of sections, please refer to **Chapter 7 — SuperReport Pro Codes (Section Codes)**.

Use Section — A valid 4th Dimension variable, which will receive the section usage setting. One of the following values may be returned

- 1 - [SR Use Section Always](#)
- 2 - [SR Use Section On Break](#)
- 3 - [SR Use Section On First Page](#)
- 4 - [SR Use Section On Second Page](#)

Print Section — A valid 4th Dimension variable, which will determine if the section is configured to be printed or not. A value of one (1) indicates the section will be printed; a value of zero (0) indicates the section will not be printed.

Position — A valid 4th Dimension variable, which will receive the vertical position of the section marker.

Options — A valid 4th Dimension variable, which will receive the print options. One of the following values may be returned.

- 1 - [SR Section Keep On One Page](#)
- 2 - [SR Section Adjust At Print Time](#)

Throw Page — A valid 4th Dimension variable, which will receive the sections “Start New Page” setting. One of the following values may be returned.

- 1 - [SR Section Throw Page None](#)
- 2 - [SR Section Throw Page Before](#)
- 3 - [SR Section Throw Page After](#)
- 4 - [SR Section Throw Page Min Space](#)

Minimum Space — A valid 4th Dimension variable, which will receive the minimum space value.

Break Type — A valid 4th Dimension variable, which will receive the section break setting, should one be set. This attribute will only be applicable when using a Subheader or Subtotal section, all other sections will return a value of zero. One of the following values may be returned.

- 1 - [SR Section Break On Field](#)
- 2 - [SR Section Break On Variable](#)
- 3 - [SR Section Break On Array](#)

Break Table Number — A valid 4th Dimension variable, which will receive sections break fields table number. This parameter is only applicable if the *BreakType* has returned a value of [SR Section Break On Field](#) (1).

Break Field Number — A valid 4th Dimension variable, which will receive sections break field number. This parameter is only applicable if the *BreakType* has returned a value of [SR Section Break On Field](#) (1).

Break Var Name — A valid 4th Dimension variable, which will receive sections break variable name. This parameter is only applicable if the *BreakType* has returned a value of [SR Section Break On Variable](#) (2) or [SR Section Break On Array](#) (3).

-> *Result Code* —If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will return all the properties for the [SR Section Body](#) section.

```
iSR_Section:=SR Section Body
iSR_UseSection:=0
iSR_PrintSection:=0
iSR_Position:=0
iSR_Options:=0
iSR_ThrowPage:=0
iSR_MinSpace:=0
iSR_BreakType:=0
iSR_BreakTableNo:=0
iSR_BreakFieldNo:=0
sSR_BreakVarName:=0
```

iSR_Err:=**SR Get Section Properties**

(vSRArea;iSR_Section;iSR_UseSection;iSR_PrintSection;iSR_Position;iSR_Options;iSR_ThrowPage;iSR_MinSpace;iSR_BreakType;iSR_BreakTableNo;iSR_BreakFieldNo;sSR_BreakVarName)

See **Also**: [SR Get Sections](#), [SR Set Section Properties](#), [SR Set Section Scripts](#).

SR Set Section Properties

SR Set Section Properties (reportArea:L; sectionID:L; useSection:L; print:L; position:L; options:L; throwPage:L; minSpace:L; breakType:L; breakTable:L; breakField:L; breakVarName:S) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Section ID	C_LONGINT	Report Section ID to obtain properties
Use Section	C_LONGINT	Use section flag
Print Section	C_LONGINT	Print section flag
Position	C_LONGINT	Section vertical position
Options	C_LONGINT	Miscellaneous Options
Throw Page	C_LONGINT	Start New Page Setting
Min Space	C_LONGINT	Minimum Space
Break Type	C_LONGINT	Section break type
Break Table	C_LONGINT	Break Table Number (when break on field)
Break Field	C_LONGINT	Break Field Number (when break on field)
Break VarName	C_STRING or C_TEXT	Break Variable Name (when break on var)
-> ResultCode	C_LONGINT	Result Code

SR Set Section Properties will assign the various properties for the given section object. If the section is not active (see **SR Set Sections**), the settings will be ignored.

Note: In most cases, passing a value of -1 will use the current value for the section. If you are unsure when to use a default value, it is recommended that you pass the desired value you wish to use.

ReportArea — The desired SuperReport Pro area.

Section ID — A valid SuperReport Pro section (i.e. SR Section Body or SR Section Header). For a complete list of sections, please refer to **Chapter 7 — SuperReport Pro Codes (Section Codes)**.

Use Section — Determines if the desired section will be used and in what capacity

- 1 - SR Use Section Always
- 2 - SR Use Section On Break
- 3 - SR Use Section On First Page
- 4 - SR Use Section On Second Page

Print Section — Determines in the desired section will be used when the report is printed.

Position — Determines the vertical position where the section is located on the report.

Options — Determines the alternate printing options available when this section is used on a printed report.

0 – No options

- 1 - SR Section Keep On One Page

Throw Page — Determines when a new page should be started (if at all) for this section.

- 1 - [SR Section Throw Page None](#)
- 2 - [SR Section Throw Page Before](#)
- 3 - [SR Section Throw Page After](#)
- 4 - [SR Section Throw Page Min Space](#)

Minimum Space — The minimum space value (in points) if the Throw Page attribute is set to [SR Section Throw Page Min Space](#).

Break Type — Determines what type of break process shall be used when this section is printed. Break processing attributes are only applicable when using a Subheader or Subtotal section.

- 1 - [SR Section Break On Field](#)
- 2 - [SR Section Break On Variable](#)
- 3 - [SR Section Break On Array](#)

Break Table Number — Determines the 4th Dimension table number of the field, which is used to determine when the page break should occur. This attribute is only required when using a *BreakType* of [SR Section Break On Field](#) (1).

Break Field Number — Determines the 4th Dimension field number of the field, which is used to determine when the page break should occur. This attribute is only required when using a *BreakType* of [SR Section Break On Field](#) (1).

Break Variable Name — Determines the variable or array that is used to determine when the break should occur. This attribute is only required when using a *BreakType* of [SR Section Break On Variable](#) (2) or [SR Section Break On Array](#) (3).

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will return all the properties for the [SR Section SubHeader1](#) section.

```
iSR_Section:=SR Section SubHeader1
iSR_UseSection:=SR Use Section Always
iSR_PrintSection:=1
iSR_Position:=-1 `use current value
iSR_Options:=SR Section Keep On One Page
iSR_ThrowPage:=SR Section Throw Page After
iSR_MinSpace:=0
iSR_BreakType:=SR Section Break On Field
iSR_BreakTableNo:=Table(->[Customer]Code)
iSR_BreakFieldNo:=Field(->[Customer]Code)
sSR_BreakVarName:=""
```

iSR_Err:=**SR Set Section Properties**

(vSRArea;iSR_Section;iSR_UseSection;iSR_PrintSection;iSR_Position;iSR_Options;iSR_ThrowPage;iSR_MinSpace;iSR_BreakType;iSR_BreakTableNo;iSR_BreakFieldNo;sSR_BreakVarName)

See Also: SR Get Sections, SR Get Section Properties, SR Get Section Scripts.

SR Get Section Scripts

SR Get Section Scripts (reportArea:L; sectionID:L; sectionScript:T; htmlPre:T; htmlPost:T) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Section ID	C_LONGINT	Report section ID
Section Script	C_TEXT	Section Script
Section HTML Start	C_TEXT	HTML Start Block
Section HTML End	C_TEXT	HTML End Block
-> ResultCode	C_LONGINT	Result Code

SR Get Section Scripts will retrieve the section script and HTML information for the defined section.

ReportArea — The desired SuperReport Pro area.

Section ID — A valid SuperReport Pro section (i.e. [SR Section Body](#) or [SR Section Header](#)). For a complete list of sections, please refer to **Chapter 7 — SuperReport Pro Codes (Section Codes)**.

Section Script — A valid 4th Dimension variable, which will receive the section script.

Section HTML Start — A valid 4th Dimension variable, which will receive the sections start HTML block.

Section HTML End — A valid 4th Dimension variable, which will receive the sections end HTML block.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will retrieve the section script and HTML data.

C_TEXT(tSR_Script;tSR_HTMLStart;tSR_ HTMLEnd)

\$SR_Err:=SR Get Section Scripts(\$SR_AREA;SR Section Body;tSR_ Script;tSR_ HTMLStart;tSR_ HTMLEnd)

See Also: SR Get Sections, SR Set Section Properties, SR Set Section Scripts.

SR Set Section Scripts

SR Set Section Scripts (reportArea:L; sectionID:L; sectionScript:T; htmlPre:T; htmlPost:T) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Section ID	C_LONGINT	Report section ID
Section Script	C_TEXT	Section Script
Section HTML Start	C_TEXT	HTML Start Block
Section HTML End	C_TEXT	HTML End Block
-> ResultCode	C_LONGINT	Result Code

SR Set Section Scripts will set the section script and HTML information for the defined section.

ReportArea — The desired SuperReport Pro area.

Section ID — A valid SuperReport Pro section (i.e. [SR Section Body](#) or [SR Section Header](#)). For a complete list of sections, please refer to **Chapter 7 — SuperReport Pro Codes (Section Codes)**.

Section Script — Desired section script for the defined section.

Section HTML Start — Desired HTML start block for the defined section.

Section HTML End — Desired HTML end block for the defined section.

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will set the section script and HTML data.

C_TEXT(tSR_Script;tSR_HTMLStart;tSR_HTMLEnd)

```
tSR_Script:="RELATE MANY([Customer])"  
tSR_HTMLStart:="<td>"  
tSR_HTMLEnd:="</td>"
```

```
$SR_Err:=SR Set Section Scripts($SR_AREA; SR Section Body ;tSR_Script;tSR_HTMLStart;  
tSR_HTMLEnd)
```

See Also: SR Set Sections, SR Set Section Properties, SR Get Section Scripts.

SR Get Tool

SR Get Tool (reportArea:L; currentTool:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Current Tool	C_LONGINT	Report tool
-> ResultCode	C_LONGINT	Result Code

SR Get Tool will return the current editor tool.

ReportArea — The desired SuperReport Pro area.

Tool — A valid 4th Dimension variable which will contain the currently selected tool in the SuperReport Pro editor.

- 0 – [SR Tool Arrow](#)
- 1 – [SR Tool Text](#)
- 2 – [SR Tool Field](#)
- 3 – [SR Tool Variable](#)
- 4 – [SR Tool Line](#)
- 5 – [SR Tool Rectangle](#)
- 6 – [SR Tool Circle](#)

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

The following example will get the currently selected tool in the SuperReport Pro editor. If the current tool is not the arrow tool, it will be set using the SR Set Tool routine.

C_LONGINT(iSR_Tool)

```
$err:=SR Get Tool($SR_AREA;iSR_Tool)
if(iSR_Tool#SR Tool Arrow)
  $err:=SR Set Tool($SR_AREA;SR Tool Arrow)
End if
```

See Also: SR Set Tool.

SR Set Tool

SR Set Tool (reportArea:L; currentTool:L) -> result:L

Parameter	Data Type	Description
ReportArea	C_LONGINT	Report plug-in area
Current Tool	C_LONGINT	Report tool
-> ResultCode	C_LONGINT	Result Code

SR Get Tool will return the current editor tool.

ReportArea — The desired SuperReport Pro area.

Tool — Desired tool you wish to set as current tool in SuperReport Pro editor.

- 0 – [SR Tool Arrow](#)
- 1 – [SR Tool Text](#)
- 2 – [SR Tool Field](#)
- 3 – [SR Tool Variable](#)
- 4 – [SR Tool Line](#)
- 5 – [SR Tool Rectangle](#)
- 6 – [SR Tool Circle](#)

-> *Result Code* — If the routine completed successfully, a value of zero will be returned; otherwise a valid SuperReport Pro result code will be returned. For a complete list of all return codes, please refer to the **SuperReport Pro Error Codes** section below.

Example

See example outlined in **SR Get Tool**.

See Also: SR Get Tool.

7 — SuperReport Pro Codes

The following chapter outlines all the SuperReport Pro Return, Error and Menu Codes. Each SuperReport Pro command will return a result, typically an error code. For more information about the various result codes, please refer to the individual commands outlined in the Command Reference.

File Manager Error Codes

One of the following error codes may be returned when using routines, which interact with disk files such as SR Load Report and SR Save Report.

Error Code	Description
-33	Directory full
-34	Volume full
-35	No such volume or volume not found
-36	I/O Error occurred
-37	Bad Filename
-38	File not open
-39	End of file; no additional data
-40	Read/ Write Error; tried to read before beginning of file
-41	Memory Full Error; file can't be opened
-42	too many files open
-43	File not found
-44	Diskette is write protected; volume is locked
-46	Volume locked
-47	File is busy (already open when trying to delete)
-48	Duplicate Filename
-49	File already open with write permissions
-51	File reference number invalid

Section Codes

When executing a routine, which requires a section code such as **SR Print Report** or **SR Set Sections**, you can use any of the following codes. If you wish to use two or more items in this list, add the numbers together to reference each item.

- To include all sections, you may pass either a value of 0, or 65535
- To include all sections except for specific sections, subtract the unwanted section(s) from 65535

Value	Constant Name	Description
65535	SR All Sections	All Sections
1	SR Section Header Mask	Header
2	SR Section SubHeader1 Mask	SubHeader1
4	SR Section SubHeader2 Mask	SubHeader2
8	SR Section SubHeader3 Mask	SubHeader3
16	SR Section SubHeader4 Mask	SubHeader4
32	SR Section SubHeader5 Mask	SubHeader5
64	SR Section SubHeader6 Mask	SubHeader6
128	SR Section Body Mask	Body
256	SR Section SubTotal6 Mask	SubTotal6
512	SR Section SubTotal5 Mask	SubTotal5
1024	SR Section SubTotal4 Mask	SubTotal4
2048	SR Section SubTotal3 Mask	SubTotal3
4096	SR Section SubTotal2 Mask	SubTotal2
8192	SR Section SubTotal1 Mask	SubTotal1
16384	SR Section Total Mask	Total
32768	SR Section Footer Mask	Footer

If you are using the **SR Get Sections** routine and would like to determine if a given section is active, you can use one of the section bit operands outlined below. In addition, you may use these constants with **SR Get/Set Section Properties** and **SR Get/Set Section** Scripts to define the sections that you wish to use.

Value	Constant Name	Description
65535	SR All Sections	All Sections
0	SR Section Header	Header
1	SR Section SubHeader1	SubHeader1
2	SR Section SubHeader2	SubHeader2
3	SR Section SubHeader3	SubHeader3
4	SR Section SubHeader4	SubHeader4
5	SR Section SubHeader5	SubHeader5
6	SR Section SubHeader6	SubHeader6
7	SR Section Body	Body
8	SR Section SubTotal6	SubTotal6
9	SR Section SubTotal5	SubTotal5
10	SR Section SubTotal4	SubTotal4
11	SR Section SubTotal3	SubTotal3
12	SR Section SubTotal2	SubTotal2
13	SR Section SubTotal1	SubTotal1
14	SR Section Total	Total
15	SR Section Footer	Footer

Error Codes

Each of the SuperReport Pro commands returns some type of result code (with the exception of the SR ABOUT routine). The following is a list of possible result codes, which may be returned.

NOTE: If the routine completed successfully, a value of zero will be returned.

Error Code	Description
-2000	SR Editor Item Invalid
-2001	SR Editor Section Not Active
-2002	SR Editor Sect Position Invalid
-2003	SR Editor Cannot Disable Sects
-2004	SR Editor Invalid Sect Option
-2005	SR Editor Incompatible Obj Type
-4001	Invalid Table (SR Structure)
-4002	Invalid Record
-4003	Invalid Field (SR Structure)
-8000	Invalid Report Format
-8001	Insufficient Memory/Fatal Error
-8002	Invalid Array Type
-8003	User Cancelled Dialog or Operation
-8004	No Selected Report Objects
-8005	Unknown Standard Variable Type
-8006	Invalid Menu ID
-8007	Invalid Parameter
-8008	Invalid Area Reference (e.g. nil, already destroyed, etc.)

In addition to the above described error codes, SuperReport Pro may return a standard Macintosh OS error code, which is not related to the internal working of SuperReport Pro. Should you receive an error code, which is not described in the above list, please refer to the **System Errors** application provided with SuperReport Pro for more details.

Event Codes

When using a custom callback method installed by **SR On Event**, one or more of the following event codes will be returned to the event method.

Event Code	Constant Name	Description
11	SR Zoom Area	Report Area Zoomed
12	SR UnZoom Area	Report Area Un-zoomed (returned to Editor)
13	SR Zoom Area to Back	Zoomed area has lost focus
14	SR Orig Area to Back	Original area has lost focus
15	SR Area Closing	Report Area has been closed
20	SR Editor Mode	Switched from Preview to Editor Mode
21	SR Preview Mode	Switched from Editor to Preview Mode
30	SR Preview First Page	Preview: First Page Button Selected
31	SR Preview Previous Page	Preview: Previous Page Button Selected
32	SR Preview Next Page	Preview: Next Page Button Selected
33	SR Preview Last Page	Preview: Last Page Button Selected
34	SR Preview Print Page	Preview: Print Button Selected
35	SR Close Preview	Preview: Close Button Selected
40	SR Menu Item Selected Before	Menu Item Selected Before
41	SR Menu item Selected After	Menu Item Selected After

Menu Numbers

When using the **SR Menu Item** or **SR Menu Info** routines, you can reference the internal number of any SuperReport Pro menu using the following table.

Menu ID	Constant Name	Description
Apple Menu		
0		<i>not displayed when valid registration number entered</i> About SuperReport Pro...
File Menu		
101	SR MenuItem New	New Report
102	SR MenuItem Open	Open from Disk
103	SR MenuItem Close	Close
104	SR MenuItem Save	Save to Disk
105	SR MenuItem Save As	Save to Disk As...
106	SR MenuItem Print To Disk	Print To Disk...
107	SR MenuItem Preview	Print Preview
108	SR MenuItem Page Setup	Page Setup...
109	SR MenuItem Print	Print...
110	SR MenuItem Zoom	Zoom Window
Edit Menu		
201	SR MenuItem Undo	Undo
202	SR MenuItem Cut	Cut
203	SR MenuItem Copy	Copy
204	SR MenuItem Paste	Paste
205	SR MenuItem Clear	Clear
206	SR MenuItem Select All	Select All
207	SR MenuItem Duplicate	Duplicate
208	SR MenuItem Modify Object	Modify Object...
209	SR MenuItem Change Object	Change Object...
210	SR MenuItem Position Object	Position Object...
211	SR MenuItem Modify Section	Modify Section
212	SR MenuItem Position Sections	Position Selections...
213	SR MenuItem Activate Section	Add/Delete Sections...
214	SR MenuItem Bring To Front	Bring To Front
215	SR MenuItem Bring Forward	Bring Forward
216	SR MenuItem Send To Back	Send To Back
217	SR MenuItem Send Backwards	Send Backwards

Options Menu

301	SR MenuItem Use Physical Page	Use Physical Page...
302	SR MenuItem Use Printable Area	Use Printable Page...
303	SR MenuItem Rulers	Show Rulers
304	SR MenuItem Ruler Units	Ruler Units
305	SR MenuItem Grid	Show Grid
306	SR MenuItem Guides	Show Guides
307	SR MenuItem Lock Guides	Lock Guides
308	SR MenuItem Sections	Show Sections
309	SR MenuItem Lock Sections	Lock Sections
310	SR MenuItem Margins	Show Margins
311	SR MenuItem Object Borders	Show Object Borders
312	SR MenuItem Show Object Border	Show Object Order
313	SR MenuItem Show Object Alias	Show Object Alias

Database Menu

401	SR MenuItem Main Table	Report/File Iterations... or Select File...
402	SR MenuItem Select Records	Select Records...
403	SR MenuItem Order Records	Sort Selection...
404	SR MenuItem Print Sections	Select Printing Sections...
405	SR MenuItem Scripts	Scripts

Modify Selection Submenu

1001	SR MenuItem Header	Header
1002	SR MenuItem SubHeader1	SubHeader1
1003	SR MenuItem SubHeader2	SubHeader2
1004	SR MenuItem SubHeader3	SubHeader3
1005	SR MenuItem SubHeader4	SubHeader4
1006	SR MenuItem SubHeader5	SubHeader5
1007	SR MenuItem SubHeader6	SubHeader6
1008	SR MenuItem Body	Body
1009	SR MenuItem SubTotal6	SubTotal6
1010	SR MenuItem SubTotal5	SubTotal5
1011	SR MenuItem SubTotal4	SubTotal4
1012	SR MenuItem SubTotal3	SubTotal3
1013	SR MenuItem SubTotal2	SubTotal2
1014	SR MenuItem SubTotal1	SubTotal1
1015	SR MenuItem Total	Total
1016	SR MenuItem Footer	Footer

Ruler Units Submenu

1101	SR MenuItem Ruler Points	Points
1102	SR MenuItem Ruler Millimeter	Mm
1103	SR MenuItem Inches	Inch
2001...n		Font List (2001.. number of fonts)
3001..3256		Color Popup for Pen
-3001..-3256		Color Popup for Fill
4001..4016		Pattern Popup for Pen
-4001..-4016		Pattern Popup for Fill

Line Thickness (Popup Menu on Report Editor)

5001	SR MenuItem Line Hair	Hairline
5002	SR MenuItem Line 1	1
5003	SR MenuItem Line 2	2
5004	SR MenuItem Line 3	3
5005	SR MenuItem Line 4	4
5006	SR MenuItem Line 5	5
5007	SR MenuItem Line 6	6
5008	SR MenuItem Line 7	7
5009	SR MenuItem Line 8	8

Font Size (Popup Menu on Report Editor)

6001	SR MenuItem Font Size 6	6
6002	SR MenuItem Font Size 7	7
6003	SR MenuItem Font Size 8	8
6004	SR MenuItem Font Size 9	9
6005	SR MenuItem Font Size 10	10
6006	SR MenuItem Font Size 11	11
6007	SR MenuItem Font Size 12	12
6008	SR MenuItem Font Size 14	14
6009	SR MenuItem Font Size 18	18
6010	SR MenuItem Font Size 24	24
6011	SR MenuItem Font Size 36	36
6012	SR MenuItem Font Size Smaller	Smaller
6013	SR MenuItem Font Size Larger	Larger
6014	SR MenuItem Font Size Other	Other...

Script (Database Submenu)

7001	SR MenuItem Start Script	Start Script Submenu
7002	SR MenuItem Body Script	Body Script Submenu
7003	SR MenuItem End Script	End Script Submenu

Editor Codes — Action Types

When using the SuperReport Pro Editor Callback interface, the callback method will receive one of the following action types (parameter 2).

Event Code	Constant Name	Description
1	SR Editor Create Object	Activated when a new object is created
2	SR Editor Modify Object	Activated when an existing object is modified
3	SR Editor Modify Section	Activated when an section is modified
4	SR Editor Modify Object Script	Activated when an existing object script is modified.
5	SR Editor Modify Report Script	Activated when one of the three (Start, Body, End) scripts is modified.
6	SR Editor Control Click Object	Activated when a control click (or Right mouse button on Windows) event occurs. This is useful when performing contextual menu access.
7	SR Editor Click Object	Activated when any object click event occurs
8	SR Editor Selection Changed	Activated when the selected object(s) has changed.

Editor Codes — Object Types

When using the SuperReport Pro Editor Callback interface, the callback method will receive one of the following object types (parameter 4).

Event Code	Constant Name	Description
1	SR Object Type Text	Text Object
2	SR Object Type Field	Field Object
3	SR Object Type Variable	Variable Object
4	SR Object Type Line	Line Object
5	SR Object Type Rectangle	Rectangle Object
6	SR Object Type Circle	Circle Object
8	SR Object Type Picture	Picture Object
1	SR Start Report Script	Report Start Script Object (when SR Editor Modify Report Script action occurs)
2	SR Body Report Script	Report Body Script Object (when SR Editor Modify Report Script action occurs)
3	SR End Report Script	Report End Script Object (when SR Editor Modify Report Script action occurs)

SuperReport Pro Constants

SuperReport Pro includes a variety of constants which can be used by each of the SuperReport Pro commands. The following table outlines all the constants, which are defined in SuperReport Pro.

Value	Constant Name	Description
1	SR Use Section Always	Section customization
2	SR Use Section On Break	Section customization
3	SR Use Section On First Page	Section customization
4	SR Use Section On Second Page	Section customization
1	SR Section Break On Field	Section customization
2	SR Section Break On Variable	Section customization
3	SR Section Break On Array	Section customization
1	SR Section Throw Page None	Section customization
2	SR Section Throw Page Before	Section customization
3	SR Section Throw Page After	Section customization
4	SR Section Throw Page Min Space	Section customization
1	SR Section Keep On One Page	Section customization
2	SR Section Adjust At Print Time	Section customization
1	SR Obj Flag Left Line	
2	SR Obj Flag Top Line	
4	SR Obj Flag Right Line	
8	SR Obj Flag Bottom Line	
15	SR Obj Flag All Lines	
16	SR Obj Flag Fixed Horizontal	
32	SR Obj Flag Fixed Vertical	
64	SR Obj Flag Grow Horizontal	
128	SR Obj Flag Grow Vertical	
256	SR Obj Flag Variable Width	
512	SR Obj Flag Variable Height	
1024	SR Obj Flag Replace If Empty	Replace Below If Empty
2048	SR Obj Flag Record Calc Value	
4096	SR Obj Flag Show Calculated Value	
8192	SR Obj Flag Repeating Object	
16384	SR Obj Flag Repeat Vertically	
32768	SR Obj Flag Repeat Horizontally	
65536	SR Obj Flag Replace Row If Empty	Replace Row Below If Empty
1	SR Variable Type Variable	
2	SR Variable Type Array Auto	
3	SR Variable Type Array Element	
0	SR Calculation Type None	
1	SR Calculation Type Total	
2	SR Calculation Type Min	
3	SR Calculation Type Average	
4	SR Calculation Type Max	

0	SR Pict Format Normal	
1	SR Pict Format Centered	
2	SR Pict Format Scaled To Fit	
3	SR Pict Format Scaled Prop	
4	SR Pict Format Scaled Prop Centered	
1	SR Iterations Main Table	
2	SR Iterations Fixed	
3	SR Iterations Variable	
4	SR Iterations Array	
-1	SR Property All	
1	SR Property Name	
2	SR Property Position	
4	SR Property Type	
8	SR Property Options	
16	SR Property Selected	
32	SR Property Field	
64	SR Property Variable Type	
128	SR Property Calculation	
256	SR Property Rows Cols	
512	SR Property Repeat Offsets	
-7169	SR Attribute All	
1	SR Attribute Font Name	
2	SR Attribute Fore Color	
4	SR Attribute Back Color	
8	SR Attribute Font Size	
16	SR Attribute Font Style	
32	SR Attribute Fore Pattern	
64	SR Attribute Back Pattern	
128	SR Attribute Justification	
256	SR Attribute Thickness	
512	SR Attribute Format	
1024	SR Attribute 4D Fore Color	
2048	SR Attribute 4D Back Color	
4096	SR Attribute No Adjust	
0	SR PowerMenu Tables	SR PowerMenu (menuID parameter)
1	SR PowerMenu Fields	SR PowerMenu (menuID parameter)
2	SR PowerMenu Variables	SR PowerMenu (menuID parameter)
3	SR PowerMenu Commands	SR PowerMenu (menuID parameter)
5	SR PowerMenu Break Object Type	SR PowerMenu (menuID parameter)
6	SR PowerMenu Variable Type	SR PowerMenu (menuID parameter)
7	SR PowerMenu Format	SR PowerMenu (menuID parameter)
11	SR PowerMenu Color	SR PowerMenu (menuID parameter)
12	SR PowerMenu Pattern	SR PowerMenu (menuID parameter)
13	SR PowerMenu Line	SR PowerMenu (menuID parameter)
-1	SR Position At Front	

-2	SR Position At End
-3	SR Position Forward
-4	SR Position Backward
0	SR All Objects
1	SR Selected Objects

8 — Hints & Tips

While SuperReport Pro has been designed to be an easy to use tool, the use of 4th Dimension library objects and plug-in routines can sometimes lead to questions that even the most seasoned 4D programmer can run into a wall. This chapter lists some of the most commonly asked questions when using SuperReport Pro.

Troubleshooting

This chapter lists commonly asked questions when using SuperReport Pro.

- Q.** Can I use SuperReport Pro in the User Environment?
- A.** Yes, SuperReport Pro can be used on any valid input form, either in the User Environment or a custom method which calls `MODIFY SELECTION`. However, SuperReport Pro does not work with the Plug-In menu item system 4D users to display various plug-in editors.
- Q.** How do I set the main table for a new or existing report that is displayed in the Report Editor?
- A.** You can use the **SR Main Table2** routine, which will work with a report area (**SR Main Table** works on the BLOB representation of the report).
- Q.** Can I use the Repeating Objects feature for multi-level relationships? For example, if I have a Customer, Invoice, and Line Item table, can I use repeating objects to show all the related Invoices, and show their related Line Items?
- A.** No, the Repeating Objects feature is designed to work with a single related many table. In order to create this type of report, you are going to have to create it based on the Line Item's table, and using Subheader data, you can show the related one and its associated related on information.

SuperReport Pro — Wrapper Routines

As an extension to the SuperReport Pro Developer API, you can create custom wrapper routines that provide users and developers with simplified access to the underlying plug-in API routines outlined above.

The SuperReport Pro demo includes a number of custom wrappers, which provide you with an example of the extensive configuration capabilities, offered by the SuperReport Pro Developer API.

Example Routine

The following is a sample routine, which enables the user to customize the appearance of a given object during printing. This routine could be called as follows:

```
SR_SetFont(SRObjectID;"";-1;1)
```

- SRObjectID is a reserved SuperReport Pro variable which contains the internal ID of the current object
- "" tells SRP to use the same font as defined in the editor
- -1 tells SRP to use the same size as defined in the editor
- tells SRP to make the current object bold

SR_SetFont(objectID:L;fontName:S;fontSize:L;fontStyle:L;justification}

`PM: SR_SetFont(objectID:L;fontName:S;fontSize:L;fontStyle:L;justification:L})

`LM: 05/22/02, mse

`\$1: objectID

`\$2: font name

`\$3: font size

`\$4: font style

`\$5: font justification

C_LONGINT(\$1;\$SR_OBJ)

C_TEXT(\$2;\$SR_FONTNAME)

C_LONGINT(\$3;\$SR_FONTSIZE)

C_LONGINT(\$4;\$SR_FONTSTYLE)

C_LONGINT(\$5;\$SR_FONTJUSTIFY)

C_LONGINT(\$SR_RET)

C_LONGINT(\$params)

\$SR_AREA:=SRArea

\$SR_OBJ:=0

\$SR_FONTNAME:=""

\$SR_FONTSIZE:=-1

\$SR_FONTSTYLE:=-1

\$SR_FONTJUSTIFY:=-1

\$params:=Count parameters

If (\$params>=1)

\$SR_OBJ:=\$1

If (\$params>=2)

\$SR_FONTNAME:=\$2

End if

If (\$params>=3)

\$SR_FONTSIZE:=\$3

End if

If (\$params>=4)

\$SR_FONTSTYLE:=\$4

End if

If (\$params>=5)

\$SR_FONTJUSTIFY:=\$5

End if

If (\$SR_OBJ=0)

\$SR_OBJ:=SRObjctID

End if

\$SR_ATTRIBS:=SR Attribute Font Name +SR Attribute Font Size +SR Attribute Font Style +SR Attribute Justification

\$SR_RET:=**SR Set Object Format**

(\$SR_AREA;\$SR_OBJ;\$SR_ATTRIBS;\$SR_FONTNAME;\$SR_FONTSIZE;\$SR_FONTSTYLE;\$SR_FONTJUSTIFY)

End if