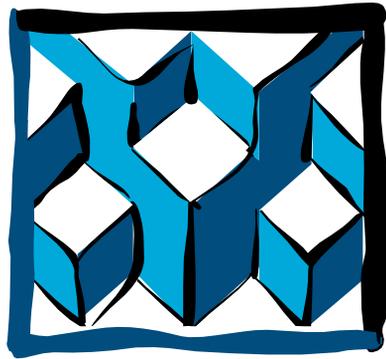


# Release Notes



# SB+ Server<sup>®</sup>

A 4GL Tool Set for Rapid Application  
Development and Guitization

*Ardent*  
Software, Inc.



All rights to this publication are reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, without prior written permission from Ardent Software, Inc. If you are a licensed user of this product, Ardent Software, Inc. grants you a limited, nontransferable license to reproduce this particular document. This is limited to photocopy reproduction of a Ardent Software-supplied master copy. Copies must include all pages contained in the master copy (including this page) and must be intended strictly for internal distribution and use by the licensee, its employees, and assigns. Reproduced copies may not be transferred, in whole or in part, to any party outside the licensee's place(s) of business. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy this software on magnetic tape, disk, or any other medium for any purpose other than the licensee's personal use.

**Copyright © 2000 Ardent Software, Inc.**  
**All Rights Reserved**  
**SB+ Server Release 5**  
**Documentation DSB-5007**  
**Rev. 02032000**

Ardent Software, Inc., reserves the right to make changes to this document and the software described herein at any time and without notice. Ardent Software, Inc. makes no warranty, express or implied, other than those contained in the terms and conditions of sale, and in no case is Ardent Software, Inc. liable for more than the license fee or purchase price of this product.

**Ardent Software, Inc.**  
50 Washington Street  
Westboro, Massachusetts USA 01581-1021  
Telephone: (508) 366-3888  
Facsimile: (508) 366-3669  
World Wide Web: [www.ardentsoftware.com](http://www.ardentsoftware.com)

Principal Technical Writer  
Tony Watkin

Contributors and Technical Reviewers  
Tim Berry, Mario Delas, Ken Lin, John McPharlin, Mark Shannon

Ardent Software is a trademark of Ardent Software Inc. uniVerse, UniData, SB+, SBOPEN, SBClient, DATA/C++, TERMULATOR, SBDesktop, ObjectCall, EasyX, DataStage, ESL, UniVerse Call Interface, UniVerse Objects, uV/Net, Retrieve, PI/open and Prime INFORMATION are trademarks of Ardent Software, Inc. TERMiTE is a trademark of Pixel Innovations, Ltd. Pick is a registered trademark of Pick Systems. Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation in the United States and or other countries. UNIX is a registered trademark in the United States and other countries, licensed through X/Open Company. All other products, standards, or company names mentioned may be trademarks or registered trademarks of their respective owners.



# Contents



---

<b>Chapter 1 - Introduction .....</b>	<b>11</b>
Overview .....	12
New Features .....	12
Modifications .....	12
New Features .....	14
HTML Output for Report Writer .....	14
HTML Output of 2nd and 3rd Level Help .....	21
HTML Process and HTML Menus .....	27
SB+ & SMART.QUERY Integration/OE Dictionary Generation .....	30
Process to Modify GUI Forms/Drivers (MODIFY.GUI) .....	36
Support for UniVerse NLS Mode .....	41
PHANTOM Process Allowing Background Processing .....	43
RedBack/SB+ Interface Demonstration .....	44
Modifications .....	47
Modified Function Key Behavior .....	47
MV Grid Control Changes .....	48
Memo GUI Object Type .....	49

Dialog Definitions Extended for GUI .....	50
Additional Process Before GUI Form Display .....	52
Other GUI Improvements .....	53
Increased User Control of Report Writer Output Selection .....	56
Default Screen Generation Extended .....	57
Modify Defaults Screen .....	58
Multi-lingual Support for Dialogs, Toolbars and Code Tables .....	59
New Paragraph Statements for OLE & VBScript .....	60
Support for Alternate Indices in the Native OE .....	62
Improved SB+ Licensing Procedure .....	63
Improved SB+ Upgrade Procedure .....	63
Extended Tool Options .....	65
Security Improvements .....	70
General Improvements .....	72

# Preface

---



## Introduction to Ardent Product Documentation

This preface contains useful information about Ardent's conventions for documenting command syntax and system output, as well as providing explanations of the specially designed icons that indicate important notes, tips, and warnings in the documentation.

### Note



This manual uses both the term SB+ and SB+ Server interchangeably with SB+ being the abbreviated expression.

---

## Features of Ardent Manuals

Ardent strives to produce useful, high quality documentation while maintaining a consistent look and feel. It will help you to know the following conventions which are common to this and all Ardent manuals. This section explains the following:

- conventions
- elements of syntax statements
- notes, warnings, and tips
- screen captures

---

## Conventions

All Ardent manuals depict command syntax according to industry standards. The following table lists the syntax conventions:

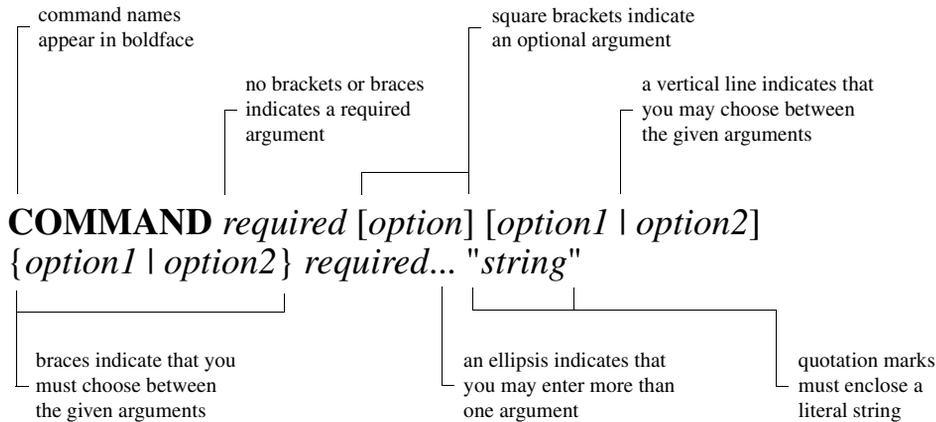
Convention	Description
<b>bold courier font</b>	Bold courier font indicates required commands that you must enter in the manner shown. You must enter all punctuation marks that appear in bold, unless otherwise indicated.
<code>courier font</code>	Courier font indicates system output, such as prompt signs, responses to commands, and program code. Courier font is also used to depict paths to directories or files.
<b>bold</b>	Bold font indicates the name of an element on the screen, such as a menu, a form, or a prompt. Also indicates the names of files or directories.
<i>lower case italics</i>	Lower case italics is used for syntactical expression of user-supplied words, variables, and expressions
<i>xx</i>	Italic lower-case xx indicates a placeholder for a system id. For example, <i>xx</i> CONTROL.
>	The > character separates each option in a menu hierarchy. For example, “choose <b>Setup</b> > <b>Miscellaneous</b> ”, means choose the Setup menu then the Miscellaneous option.
[ ]	Square brackets indicate an item that is optional. You may include a number of items enclosed in brackets in a Unidata command or function.
{ }	Braces indicate that you must choose one of the items separated by the vertical bar ( ).
	Vertical bar separates option arguments from which you may choose.
...	An ellipsis (...) after an argument indicates that you may use more than one argument on a single command line.

### Conventions

---

## Elements of Syntax Statements

SB+ Server has many commands that you enter as part of paragraphs. The syntax statement includes the command name, required arguments, and options that you can use with the command. Italics represents a variable that you can replace with any valid option. The following figure illustrates the elements of a syntax statement:



## Important Notes, Warnings, and Tips

You will notice several different icons throughout this manual. These icons draw attention to important information about the product.

### Note

---

A **Note** icon indicates important additional information on the subject.

---

### Warning

---

A **Warning** icon alerts you to the danger of deleting or corrupting data.

---

---

**Reminder** 

A **Reminder** icon marks information that is discussed in detail in another chapter or another Ardent manual.

---

**Tip** 

A **Tip** icon denotes shortcuts, commands, or procedures that may help you use Ardent products more efficiently.

---

# ***Chapter 1 - Introduction***

---



This chapter provides an overview of the new features and enhancements that have been made in this release of SB+ Server.

## Overview

Listed below is a summary of the enhancements to release 5 of SB+ Server.

### New Features

- HTML Output for Report Writer
- HTML Output of 2nd and 3rd Level Help
- HTML Process and HTML Menus
- SB+ & SMART.QUERY Integration and OE Dictionary Generation
- Process to modify GUI forms/drivers (MODIFY.GUI)
- Support for UniVerse NLS mode
- PHANTOM process allowing Background processing
- RedBack / SB+ Interface Demonstration

### Modifications

- Modified function key behavior
- MV Grid Control changes
- Memo GUI object type
- Dialog Definitions Extended for GUI
- Additional Process Before GUI Form Display
- Other GUI Improvements
- Increased User Control of Report Writer Output Selection
- Default Screen Generation Extended
- Multi-lingual support for Dialogs, Toolbars and Code Tables
- New Paragraph statements for OLE & VBScript

- Support for Alternate Indices in the Native OE
- Improved SB+ Licensing Procedure
- Improved SB+ Upgrade Procedure
- Extended Tool Options
- Security Improvements
- General Improvements

## New Features

### HTML Output for Report Writer

A new HTML output option is available in release 5 of SB+ in GUI mode. This feature may be used as an alternative to the existing Report Writer GUI output to a CEO.

The F6 **Report Parameters** screen contains a new prompt **Allow HTML** alongside the **Output Device** prompt.

Multiple Records/page	<input type="text" value="Y"/> (N/Y/H)		
Description Source	<input type="text" value="C"/> (N/S/A/C)		
Output Device	<input type="text" value="A"/> (S/P/X/D/F/A)	Allow HTML	<input type="text" value="Y"/> (Y/N)
Totals Only	<input type="text" value="N"/> (Y/N/A)	Align Forms	<input type="text" value="N"/> (Y/N)
Two Pass Report	<input type="text" value="N"/> (Y/N)		
Screen Window Co-ords	<input type="text"/>		
Sort Fields	<input type="text"/>		
Break Fields	<input type="text"/>		
Selection Criteria	<input type="text"/>		
Process At Start	<input type="text"/>		
Process After Read	<input type="text"/>		
Process Before Break	<input type="text"/>		
Process After Break	<input type="text"/>		
Process At End	<input type="text"/>		
Pages to Store	<input type="text" value="10"/>		

F2-Accept   F5-Gen Deft   F6-Addit   F7-Convert Query   F8-Convert Screen   F9-Colors

If **Output Device** is set to 'A' and **Allow HTML** is set to 'Y', the **Output Direction** dialog will contain an HTML check-box. This may be checked to output the report in HTML format to the screen, the auxiliary printer or to file.



If **Output Device** is set to 'S', 'X' or 'F' and **Allow HTML** is set to 'Y', the report will be output in HTML format directly to the screen, the auxiliary printer or to file.

When output is directed to the screen, an HTML report template driver is generated from the existing Report Writer drivers. The template is populated with report data at runtime and the resultant HTML report is transferred to the client and displayed in the Microsoft Internet Explorer Web Browser ActiveX control.

### Note

HTML has been implemented effectively as a "style" within the existing output options. Thus when HTML is being output to the screen, @RV.OUT.DEVICE will still show "S". To determine whether this "screen" output is in fact to the screen or to HTML within an ActiveX window, there is now an additional @RV variable, @RV.HTML, which will be set to 0 (false) if the output is to the standard CEO provided by SB+.

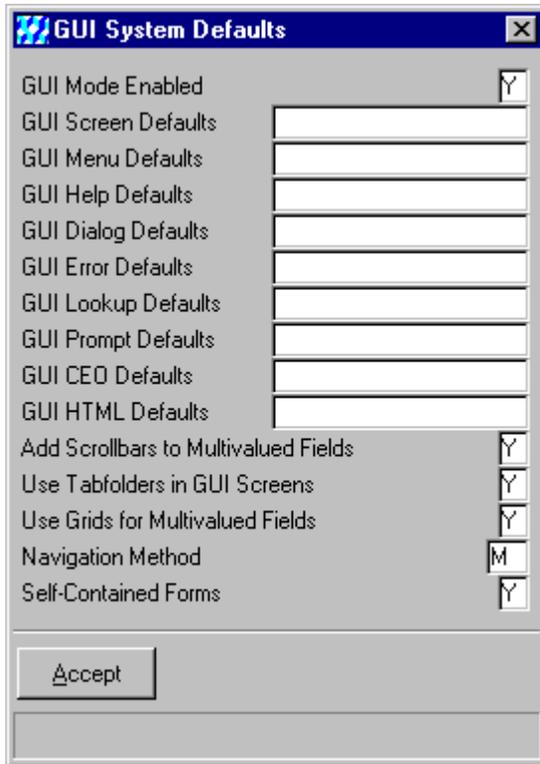
Non zero values signify:

- 1 - output is to HTML in a browser control window
- 2 - output is to Windows printer using a browser control
- 3 - output is to a Windows file using a browser control

A new GUI Defaults Set 'HTML.GUIDEFS' is available to modify the look of HTML output reports, see **Technical Help > GUI Features > GUI Presentation Defaults > Presentation Item** in the General Help menu.

There are three new fields relating to HTML.GUIDEFS in SB+ screens:

1. In the SB+ admin account, there is a new field 'GUI HTML Defaults'. This is a default setting applied to new accounts when they are created. It does not directly control any SB+ actions. This field is on the **GUI System Defaults** F6 subscreen, which is reached via the F5 **System Defaults** subscreen from the **SB+ Control Parameters** screen.



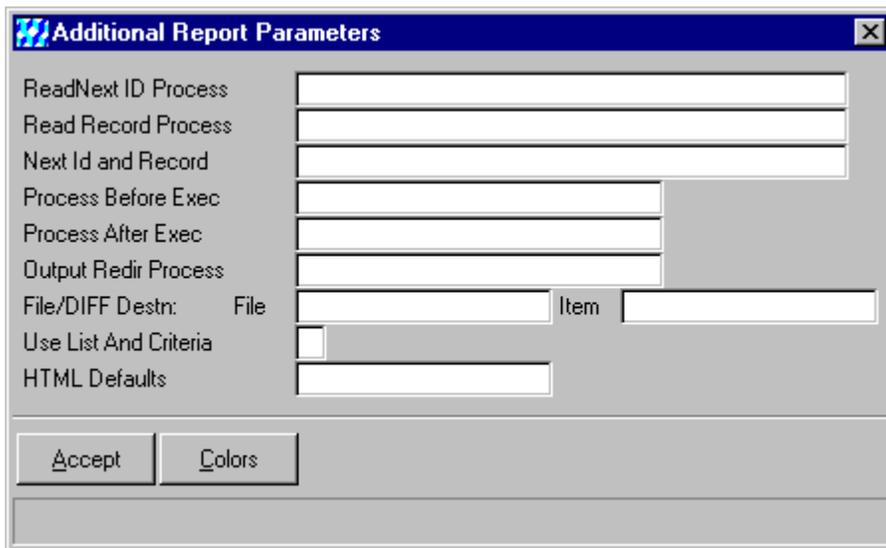
The screenshot shows a dialog box titled "GUI System Defaults" with a close button (X) in the top right corner. The dialog contains several settings:

- GUI Mode Enabled:  Y
- GUI Screen Defaults:
- GUI Menu Defaults:
- GUI Help Defaults:
- GUI Dialog Defaults:
- GUI Error Defaults:
- GUI Lookup Defaults:
- GUI Prompt Defaults:
- GUI CEO Defaults:
- GUI HTML Defaults:
- Add Scrollbars to Multivalued Fields:  Y
- Use Tabfolders in GUI Screens:  Y
- Use Grids for Multivalued Fields:  Y
- Navigation Method:  M
- Self-Contained Forms:  Y

At the bottom left, there is an "Accept" button. Below the button is a horizontal scrollbar.

2. For each module of each account, in the **GUI System Parameters** subscreen (F6) of the **System Control Record** screen, there is a new field also labeled **GUI HTML Defaults**. This record controls the characteristics of the report writer HTML output. When a new module is created, this field will default to the same value as the field in the SB+ admin account.
3. For each report, the module default can be overridden by specifying a different HTML record at the **HTML Template** field in the **Additional Report Parameters** F6 subscreen, which is reached via the F6 **Report Parameters** screen in the RD tool. If this field is blank and no

default has been specified for the module in the **GUI System Parameters** subscreen of the **System Control Record** screen, the item HTML.GUIDEFS will be used.



The screenshot shows a dialog box titled "Additional Report Parameters". It contains the following fields and controls:

- ReadNext ID Process: [Text Input]
- Read Record Process: [Text Input]
- Next Id and Record: [Text Input]
- Process Before Exec: [Text Input]
- Process After Exec: [Text Input]
- Output Redir Process: [Text Input]
- File/DIFF Destr: File [Text Input] Item [Text Input]
- Use List And Criteria:
- HTML Defaults: [Text Input]

Buttons at the bottom: Accept, Colors.

## HTML Report Driver Template Editing

A mechanism for editing the HTML report driver template is available to modify the look of the report for HTML display, for example to remove fields not relevant to HTML output such as Page Number, or to remove graphics that might clutter the new format. When running a report output in HTML format, SB+ checks to see whether a template with the same name exists already in either the module or global definition file. If one is found in either location (and they are checked in the order: global definition file, then local xxDEFN file), it is used instead of generating a new template. Therefore, the standard template generated by running the report in HTML format can be downloaded to the client, modified and then uploaded to the module or global definition file, to provide a look tailored to HTML.

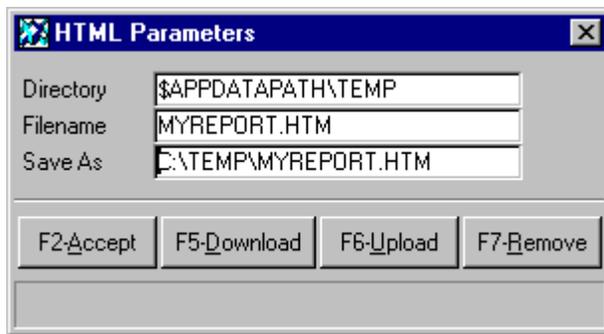
### Note



Once a template exists, it does not get updated automatically when changes are made to the underlying report - either these changes must be made manually to the template or else the template must be regenerated from the report.

An HTML format report derived from a user modified template will have '(U)' appended to the title of the report displayed in the title bar of the form containing the Active X Web Browser control used to display the report.

Attached to the Report Writer Process definition is a new subscreen F5>HTML.



The sub-screen contains prompts for a DOS Directory on the client, which may include SBClient environment variables, and a DOS Filename.

The **F5>Download** function key will download the HTML report driver template to the specified location for editing using, for example, an installed HTML editor.

The **F6>Upload** function key will upload the edited HTML report driver template back to the host to the system definition file `xxDEFN`, or to the Global definition file if one is specified on the System Control Record. Subsequent report output to HTML will use the edited HTML report driver template from the definition file.

The **F7>Remove** function key will remove the edited HTML report driver template from the definition file and subsequent report output to HTML will revert back to the system driver file `xxDRIVERS`. This process may be accomplished manually on the host, by simply copying a modified HTML report driver template to the system definition file.

The HTML report template will contain markers to indicate the end of one report section and the start of the next, eg. At the end of the detail section will be an HTML marker tag, '!-- D.END--'. Any additional HTML code for the detail section should be inserted above this line. An end of section marker will be present within the HTML template for each report section used in the report; Heading, Detail, Break, Grand Total and Footing sections.

A location may be specified at the **Save As** prompt to which the report will be saved when HTML report output is directed to File. This location will also be used as a default when saving the report from the File menu of the Web Browser Control used to display the report.

## Image Report Field Type

To take advantage of some of the features available in HTML, images may be defined within the Report Writer definition for display when output to HTML. A report Field Type 'I' may be used to resolve a DOS file path to an image file on the client. The DOS file path may be a Report Definition derived field, a field definition derived field or a field data value. The image fields will be processed only when output is directed to HTML. SBClient environment variables may be used in the DOS file path.

## Performance Issues

Using the HTML option with large reports may produce an unacceptably slow response from the Web Browser Control. The HTML.GUIDEFS GUI presentation default item contains an attribute 'display\_lines' which may be set to break up the runtime HTML code into more manageable sections for the Web Browser Control. For example, setting 'display\_lines' to 50 will split the runtime HTML report code into sequential sections of 50 lines of report data.

A mechanism for paging will be appended to each page of HTML code in the following format:

```
Start Prev 1 2 3 4 5 6 Next End
```

## Printing

The currently displayed section of the HTML report may be output to a Windows printer using the 'Print' option on the File Menu of the Web Browser Control window.

The entire HTML report may be output to a Windows printer using the 'Print All' option on the File Menu of the Web Browser Control window. An HTML report split using the 'display\_lines' attribute of the HTML.GUIDEFS GUI presentation default item will be pieced together for printing, ie. each section will NOT be printed individually.

## Saving an HTML Output Report

The HTML output report may be saved to the client using the 'Save As' option on the File Menu of the Web Browser Control window. The 'Save As' dialog will prompt for a File Destination, defaulting to the location specified at the 'Save As' prompt of the 'HTML Parameters' subscreen of the

Report Process Definition screen. An HTML report split using the 'display\_lines' attribute of the HTML.GUIDEFS GUI presentation default item will be saved as a series of pages with the page number appended to filename.

## Calling an SB+ Process from the HTML Output Report

A new field **HTML Process** has been added to the **Field Definitions > F6 Additional Field Parameters** subscreen and to the Report Writer Definition **Enter Field** subscreen. An HTML process specified in the Report Writer Definition will override an HTML process specified in the Field Definition.

A comma delimited list of parameters after the process name will be passed to the SB+ process in the common variable @PARAM.

A space delimited list of parameters after the process name will be data stacked to the SB+ process.

The actual run-time data value of the field may be used in either list using the reserved word marker @DATA.

eg. HTML Process    I\*CUSTOMER\*SCR1 @DATA

will data stack the actual run-time data value of the field to the SB+ process I\*CUSTOMER\*SCR1

eg. HTML Process    CALCULATE.PROFIT,17.5,@DATA,6

will pass the parameters 17.5, the actual run-time data value of the field, and 6 to the SB+ process CALCULATE.PROFIT in the common variable @PARAM.

## **HTML Output of 2nd and 3rd Level Help**

Second and third level of SB+ help is now converted to and displayed as HTML in GUI mode. No special conversion of existing help text is required.

### **Display Modes**

There are now two Help Display modes:

#### **Text**

Help will be displayed in 'TEXT' mode if the Help Type (attribute one) contains <TEXT> or in Character mode.

#### **HTML**

Help will be displayed in 'HTML' mode if the Help Type (attribute one) does not contain <TEXT> and current mode is GUI.

### **Conversion of Text**

SB+ will automatically convert headings, points and tables if it can recognize the format. It will also create 'internal' links from points to headings.

Text is converted as follows:

## Headings

**Heading**   **Heading**   **Heading~**  
=====   -----

*Converted to:*  
**Heading**

## Points

**\* Point**   **+ Point**   **- Point**

*Converted to:*

- Point

## Tables

```
+-----+-----+
| Column 1 | Column 2 |
+-----+-----+
| Text 1-1 | Text 1-2 |
| Text 1-2 | Text 2-2 |
+-----+-----+
```

*Converted to:*

Column 1	Column 2
Text 1-1	Text 1-2
Text 1-2	Text 2-2

## HTML Help Types and Modifiers

Attribute one of the Help item may contain a type and multiple modifiers.

## Types

One of the following types may be used:

- `<TEXT>` - display help in 'TEXT' mode
- `<HTML>` - text already converted to HTML, display in 'HTML' mode if possible
- `<EHTML>` - text contains 'embedded' HTML, display in 'HTML' mode if possible
- None of the above - display in 'HTML' mode if possible

## Modifiers

One or more of the following modifiers may be used with types `<HTML>` and `<EHTML>`:

- `<ID=filename>` - specify the filename of the generated HTML
- `<TITLE=title>` - nominate the title of the Help Form
- `<CD=xc,yc,xd,yd>` - coordinates and dimensions of the Help Form
  - xc - X coordinate
  - yc - Y coordinate
  - xd - X dimension
  - yd - Y dimension
- `<LINKS>` - check all 'external' links (links to other files)

## Display Help in 'TEXT' mode (`<TEXT>`)

Help item is not converted or displayed as HTML.

## Embedded HTML (`<EHTML>`)

If the first line of a Help item contains `<EHTML>`, it may contain 'embedded' HTML.

A sub-set of HTML commands are supported, which are acted upon in 'HTML' mode and removed in 'TEXT' mode.

All lines will be joined together, except if:

- there is a blank (NULL) line in-between them
- the previous line is a Heading, Point or Table
- the previous line ends with <P> or <BR>
- current mode is 'TEXT'

### Available HTML Commands

<TEXT>Text</TEXT>

Text will only be displayed in 'TEXT' mode.

<HTML>Text</HTML>

Text will only be displayed in 'HTML' mode.

<P>

New paragraph

<BR>

New line (lines which do not end with <P> or <BR> will be joined together)

<A HREF="#reference">Link</A>

A Link to an Anchor

<A NAME="reference"><A>Anchor</A>

An Anchor

<PRE>Preformatted Text</PRE>

Preformatted Text

<B>Bold</B>

Bold

```
<EM>Emphasized</EM>
```

Emphasized

```
<TABLE BORDER=1>
<TR><TH>Column 1</TH><TH>Column 2</TH></TR>
<TR><TD>Text 1-1</TD><TD>Text 1-2</TD></TR>
<TR><TD>Text 2-1</TD><TD>Text 2-2</TD></TR>
</TABLE>
```

Column 1	Column 2
Text 1-1	Text 1-2
Text 2-1	Text 2-2

```
<HR>
```

Horizontal line

## Note

To force any external links (`<A HREF="path/filename">Text</A>`) to be checked at runtime and removed if not found, append `<LINKS>` to attribute one (after `<EHTML>`).

## HTML (`<HTML>`)

If the first line of a Help item contains `<HTML>`, the text will NOT be converted to HTML.

This type can only be used for Help items which are already converted to HTML.

## Calling Help

Links can be included to call SB+ Help within the same Browser. In other words, further levels of help can be displayed within the same window.

The link is specified as follows:

```
<A HREF="#%HELPIITEMID%">Help Description</A>
```

## Program Example



```
<A HREF="#"%CUSTOMER3%">More</A>
```

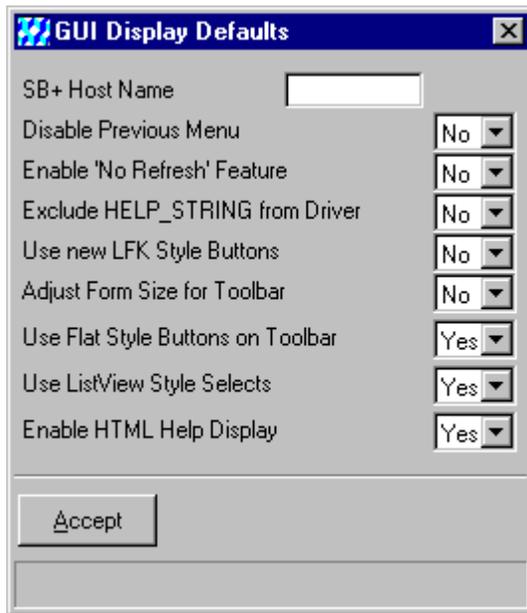
---

## Disabling HTML Help

HTML help display mode is enabled by default. It is possible to always display Help in standard 'TEXT' mode, by modifying a system-wide **Enable HTML Help Display** setting via SB.PARMS process.

### SB.PARMS > F8-Disp Defaults > F5-GUI Display Defaults

Enter the required setting into the Enable HTML Help Display field.



---

## HTML Process and HTML Menus

Apart from standard SB+ help items, other 'non-help' text can be displayed in HTML style via the new HTML process. HTML style can also be used to build and display menus, linking SB+ Processes to HTML text.

### HTML Process

The Tool Process HTML allows HTML to be displayed from within SB+.

#### Syntax

HTML,[*filename*,]*itemid*

*filename* - optional filename, if not specified *xxHELP* will be assumed.

*itemid* - id of the HTML item.

#### Program Example



HTML, XXWORK, HTML1

---

#### Program Example



HTML, HTML2

---

## Conversion of Text Into HTML

Text will still be converted into HTML, as with Help. The HTML process also allows for both embedded and non-embedded HTML.

### Client HTML

HTML will also display a HTML file which only exists on the Client. If the specified itemid does not exist on the Server, it assumes that it exists on the Client and acts accordingly.

### Program Example



HTML, HTML1 . HTM

---

The itemid can be specified as the full path or just the filename. If only the filename is specified it assumes \$SBTUPATH/temp/filename.

### Web Links

To display a Web site, a URL can be specified for itemid.

### Program Example



HTML, www.ardentsoftware.com

---

### Program Example



HTML, http://www.ardentsoftware.com

---

### Calling Processes

Links can be included to call SB+ Server Processes. In other words, a Process may be called from HTML Help.

The link is specified as follows:

`<A HREF="#"$PROCESSNAME$">Process Description</A>`

---

**Program Example**

```
<A HREF="#"#$CUST.MAINT$">Customer Maintenance</A>
```

---

**HTML Menus**

The HTML process can be used to create HTML menus

**Program Example**

```
<EHTML><ID=CUSTOMENU><TITLE=Customer Menu><CD=0,0,250,250><LINKS>  
<B>Customer Menu</B><P>  
<A HREF="#"#$CUST.ENTRY$">Customer Entry</A><BR>  
<A HREF="#"#$CUST.DISP$">Customer Display</A><BR>  
<A HREF="#"#$CUST.REP$">Customer Report</A><BR>  
<A HREF="http://www.ardentsoftware.com">Ardent Software</A><BR>
```

---

## SB+ & SMART.QUERY Integration/OE Dictionary Generation

The SBClient SMART.QUERY utility originally started out with a view to providing ad-hoc reporting tool outside scope of SB+. In some areas SMART.QUERY overlapped with SB+ Query Reports. Subsequently, the functionality of SMART.QUERY (particularly its range of output options) has come to exceed that of the SB+ Query tool. This has led to an enhancement request from SB+ users to have this functionality available to them in SB+. At the same time, users of SMART.QUERY who also have SB+ installed find that it is unnecessarily limited, because it does not take advantage of the greater power of the extended SB+ dictionary definitions.

For SB+ release 5, substantial effort has been made to satisfy the demands of SMART.QUERY users to allow SMART.QUERY to "understand" SB+ dictionary definitions. At the same time, the demands of SB+ users to allow the SB+ Query tool to provide the same output functionality as SMART.QUERY have been addressed. As a side effect of this, release 5 features two other significant enhancements to SB+:

1. The creation of OE dictionary definitions for all SB+ expressions (currently, only the simplest of SB+ expressions are passed onto the OE dictionary definition as correlatives - all others are created as "ERR"); and hence
2. The ability to use dictionary definitions that employ SB+ expressions as selection criteria. Prior to this release, SB+ expressions that cannot be passed onto an OE dictionary definition cannot be used as selection criteria, since SB+ generally relies on the OE selection processor (unless the selection is on a single criteria which is also a BTree field for that file).

In instances where SB+ would previously have been unable to convert its internal expressions into an OE correlative, it will now insert a call to a new subroutine (SB.SMART.QUERY). This subroutine acts as an interface between SB+ and the OE, allowing the expressions to be evaluated and resolved by SB+ itself. The result is then returned to the OE Query processor in the (non-SB+) form it expects. This means that the host library SMART.QUERY program has not had to change at all - it does not have to be "aware" of SB+ because the dictionary definition items themselves are

able to communicate directly with SB+. Activation of this new feature is controlled by a new flag "SB+ Subroutines in Dictionaries" in the **System Defaults** subscreen of the **SB+ Control Parameters** process.

Default Date Conversion	D4/	Accept
Default Money Conversion	MR22	Additional
Process At Set Common		GUI
Process At OE Return		Language Table
Process Before CHAINing		NLS Support
Field Template Dict		
Global Dict Name		
Enable Multi-language Support	Yes	
SB+ Subroutines in Dictionaries	Yes	

### Note



This functionality is available on the following platforms: UniData, UniVerse and D3.

D3 a separate dictionary subroutine is generated for each dictionary item. These subroutines are stored in the *xxPROGS* file of the module to which the file belongs. A new process "CATALOG.SBSQ" has also been provided which allows you to (re)catalog all SB+ generated subroutines which are called from the dictionary of the file. This process can be used after setting a "Q" pointer to the file from a remote account.

The report definition records created and used by the host library SMART.QUERY program are very similar to those created and used by the SB+ Query tool. Aside from a few of the fields in the definition records being in a slightly different order, there is no significant difference that can not be overcome "on the fly". For example, SMART.QUERY stores the output destination in the record, while the SB+ Query tool generally prompts for it at run time. This means that the existing SB+ Query tool can now simply offer the additional SMART.QUERY output options along with its current output options. If one of the SMART.QUERY options is selected, SB+ Query tool can then change its query definition record to SMART.QUERY format and make a direct call to the same routine that is used by SMART.QUERY.

The handling of output destinations for SB+ Query reports is now allocated as follows:

- i. These destinations require a knowledge of SB+ (eg terminal definitions, SB+ Print Manager) and will continue to be handled within SB+:
  - "Ask";
  - Background;
  - Screen (character);
  - Labels;
  - Host Printer;
  - Auxiliary Printer.
- ii. These destinations were not previously available from within SB+ but can now be provided to SB+ Query reports by SMART.QUERY (SBClient is required for these destinations):
  - GUI Screen;
  - Excel;
  - Lotus 123;
  - Word.
- iii. These destinations are not currently available via SMART.QUERY and so will continue to be handled within SB+ (via the existing SBLIST/SBSORT interface between the SB+ Query tool and the SB+ Report Writer) until similar functionality is provided through the new common interface (see SBClient documentation):
  - DIF file;
  - OE File;
  - HTML.
- iv. These file destinations/formats are presently not available within SB+ or via SMART.QUERY, but are available via the HOST2PC host library routines and have been made available to SB+ via the new common interface in the SBClient host library:
- v. Fixed Length;

- vi. Comma Separated Variable;
- vii. Tab Delimited;
- viii. Space Delimited;
- ix. ASCII;
- x. BINARY;
- xi. HEX;
- xii. dBase;
- xiii. SYLK.

Once SMART.QUERY definitions have been created, users can run them from SB+ menus without having to bring up the SMART.QUERY form at all. That can be achieved via new process type ("Q") which allows users to link host based (TUQUERY file) SMART.QUERY reports to SB+ menu options, function key buttons, etc in the same way that SB+ Query reports (process type "E") are handled now.

In fact, once the SMART.QUERY record is reformatted into the form that the SB+ Query tool uses, the processing is identical.

Users can also convert SMART.QUERY definitions to SB+ Query definitions permanently. This gives them access to the "Proc Before Exec", "Proc After Exec" and "Other Options" features available to SB+ Query reports, but not available to SMART.QUERY reports. In this instance, the SMART.QUERY definition continues to exist in TUQUERY, but a separate SB+ Query definition is built from it and written to xxDEFN. This is achieved via new function keys in the **Query Report Defns** screen: "F6-Host SMART.QUERY" and "F7-Client SMART.QUERY", similar to the renamed "F7-Convert Query" function key used for converting SB+ Query definitions to SB+ Report Writer definitions in the SB+ Report Writer tool (see also the "[SB+ Function Key Names Standardized](#)" on page 72).

**Query Report Definitions**

Review Print Copy Help

Report Name: QUERY.ONE  
Report Description: QUERY REPORT  
Dict File Name: SBFEEDBACK  
Data File (If diff):  
Sort Fields:  
Fields To Print: ( 20 Chars)  
FEEDBACK.NO SBD.WHO.FIXED  
Selection Criteria:  
PRODUCT = "SBC" AND DATE.FIXED > "01/01/99"  
Output Destination: S File:  
Totals Only (Y/N): N Item:  
Double Spacing (Y/N): N  
Select List Name:  
Label Parameters:  
Preferred Heading:  
Footing:  
Save Del Host SMART.Query Client SMART.Query Addit

Since SMART.QUERY creates separate work files for each port (TUXFER.DATA.nn), a new SB+ process, CLEAN.SBSQ which is similar to the existing CLEAN.DICTS, CLEAN.SBHSTATE and CLEAN.SYSTEM, has been provided to allow users to execute the SBClient host program TU.CLEAN, allowing these files to be easily removed.

These enhancements rely on the use of Named COMMON block and the ability of the underlying OE platform to support calls from dictionary definition items to subroutines which then can be interfaced to the standard SB+ environment.

Currently, this means that these features are available only on the following platforms: Universe, Unidata and D3.

Existing SB+ Query definitions can be run through the more sophisticated SB+ Report Writer tool "on the fly" (by setting the "Use SBSORT/SBLIST" flag) and can also be permanently converted into Report Writer definitions. Once they become Report Writer definitions, the report structure and output layout has the potential to become considerably more complex. Please note that the SMART.QUERY features now made available to the SB+ Query tool have not been extended to the Report Writer tool.

Please note also that as SB+ does not currently create even simple report definitions for SQLator/Oracle where these would involve creation of temporary dictionary items (these have to be created by SQLator's own tools), the new functionality has not been provided for versions of SB+ using SQLator to interface with Oracle files.

## Process to Modify GUI Forms/Drivers (MODIFY.GUI)

### Process: MODIFY.GUI

A new process has been created to complement the existing REGEN.GUI and REGEN.GDS processes. This new process, called MODIFY.GUI, allows the user to change the specific object attribute (including the attributes of the form object itself) on any number of forms at the same time. This way there is no need to repeat identical changes to each form in the form painter or completely regenerate the forms using REGEN.GUI.

Object Class Name	Attribute	New Value
textclass	background	255;0;0

Process File: GLPROCESS

Process Names	Description	Stat
DOS.SETUP	Dos Demo Setup	G

Options: 0

Accept

In a manner similar to REGEN.GDS (and both MODIFY.GUI and REGEN.GDS differ from REGEN.GUI in this respect) MODIFY.GUI does not rebuild the ".GUI" form definition record from the initial character screen definition, but updates the specified attributes. However MODIFY.GUI does differ from REGEN.GDS in three ways:

1. MODIFY.GUI does not restrict the user to only making changes to attributes which can be defined in a GDS (GUI Default Set) record - it accepts any attribute which can be modified in the properties window of the form painter.
2. MODIFY.GUI does not require all forms to be using the same GDS record.
3. MODIFY.GUI is only designed to modify one object class/attribute at a time.

The syntax for calling MODIFY.GUI is:

```
/MODIFY.GUI[,[options],[process_filename],object_class,attribute[,new_value]]
```

### options

These options are the same as those available at the Options prompt in the Modify GUI Definitions screen. See the 'Options' prompt below. If no options are specified, the default is D.

### process\_filename

You can specify a process file other than the default file (*xxPROCESS*).

### Note



Some screen processes require data in the *xxDEFN* file, as well as the *xxPROCESS* file. If you specify the global process file, MODIFY.GUI attempts to open the global Defn file. If this fails, or no global Defn filename is found in the system control record, MODIFY.GUI attempts to derive the name of the associated Defn file from the name of the process file. If this fails too, MODIFY.GUI reports an error and quits.

The MODIFY.GUI process supports active select lists (similar to the active select list feature in similar processes such as Generate Source Code (/GC), REGEN.GUI, REGEN.GDS and VALIDATE.GUI. To use an active select list, call the MODIFY.GUI process after generating a select list of process names. For example:

```
EXEC 'S:SSELECT GLPROCESS WITH TYPE = "I" (S'  
EXEC 'MODIFY.GUI,DQ,GLPROCESS,SCREEN.GUIDEFS'
```

### **object\_class**

The name of the object class whose attribute to be changed. This is mandatory. See the 'Object Class Name' prompt below.

### **attribute**

The name of the attribute to be changed. This is mandatory. See the **Attribute** prompt below.

### **new\_value**

The value to be used for this attribute. May be null. See the **New Value** prompt below.

## **Function Keys**

### **F2 Accept**

Accepts the Modify GUI screen.

## **Prompts**

### **Object Class Name**

Enter the name of the object class whose attribute you wish to change. This can be any object which you can paint onto a form using the form painter. There is F3 intuitive help available.

### **Attribute**

Enter the name of the attribute you wish to change. This can be any attribute which can be modified in the properties window of the form painter. There is F3 intuitive help available.

### **New Value**

Enter the value to be used for this attribute. This may be null if you wish to force the form to use the Windows default for this attribute.

Note these special considerations:

1. For attributes which have multiple values delimited by semicolons (eg. coordinates, dimensions), a question mark ("?") can be used to leave the value of that position as it is. For example, supplying:

?;2

as a dimensions value would leave all x-dimensions unchanged, but would cause all y-dimensions to be set to 2.

2. When modifying the dimensions of an object which can have a "direction" (presently only radioclass and separatorclass), the values entered are taken to be for the horizontal direction. As each object is updated, its direction is tested and if found to be vertical, the specification is reversed. That means that supplying:

?;2

as a dimensions value for separatorclass objects would cause all horizontal separator lines be reset to a depth of 2 and all vertical separator lines be reset to a width of 2 (while leaving the other dimensions unchanged).

3. "WHITE.SPACE" and "TEXT.FACTOR" are treated as special cases -
  - specifying them as null will reset them to their predefined default values - it will not reset them to null; and
  - question marks cannot be used

F3 intuitive help for this field is valid for some attribute types.

### **Process File**

The process file name. This allows you to specify another file name other than the default file (*xxPROCESS*).

### **Process Names**

The names of the screen processes to update.

F3 intuitive help allows you to select one of the following:

- all screen processes
- screen processes that do not have a GUI form definition
- screen processes that have a GUI form definition, but have not been modified with the SB+ Form Painter.

The selection criteria is further restricted to one of the following process types:

H Shell

I Input

O Output

T Transaction

### Description

Displays the screen process description. This information is entered in the Description prompt of the process definition (display only).

### Stat

Displays the status of the process (display only). Status is one of the following:

G the process has a GUI form definition

M the process has a GUI form definition, that has been modified in SB+ Form Painter

### Note



---

If a process does not have a GUI form definition then it cannot be updated by MODIFY.GUI. REGEN.GUI should be used instead.

---

### Options

D Forces the screen driver to be rebuilt (default)

R Forces existing source code to be Regenerated

Q Quiet mode. No progress messages are displayed on the status line

T Transfers the GUI form definition to the client PC

### Note



---

The "R" option only takes effect if the D option is also set to force a new driver to be generated and source code has been generated previously.

---

## Support for UniVerse NLS Mode

NLS (National Language Support) is a program's ability to adapt to languages and national conventions required by users from all over the world. UniVerse has a special mode that offer this support. This release of SB+ takes advantage of this feature and allows SB+ applications to be internationalized easily (currently the NLS support in SB+ is limited to characters with single width display.)

This document only details the aspects of NLS mode relevant to SB+. Consult the UniVerse documentation for issues related to NLS implementation and database setup.

To ensure that SB+ runs correctly in UniVerse NLS mode, the following steps should be carried out first. Do a backup of your system before proceeding.

- Determine the required file mapping for DMCONT (if required). This file will contain text pertaining to users' locale settings. In normal situations there should be no specific mapping required. If there will be users using a variety of locales, then it is probably best to use an unmapped file.
- Run NLS.UPDATE.ACCOUNT in each existing SB+ accounts, including the SB+ Admin account. This will convert the character set mapping for some UniVerse files. If errors occurs, it may be necessary to modify or delete some data that the command cannot convert, then try again.
- There is a new subscreen in SB+ Control Parameters which allows field conversion codes to be mapped automatically at run-time. You may need to enter the appropriate values here if you decide to utilize this feature. See the explanations below for more details.

Once SB+ is running in UniVerse NLS mode, you can start developing new applications that will run in this environment. You can also convert existing applications to utilize this support. Either way, you should note the following points:

- When creating new date, time and monetary fields, you may choose to use conversion codes that are NLS specific. They are 'DI', 'TI' and 'MM' for date, time and monetary fields, respectively.

### Note



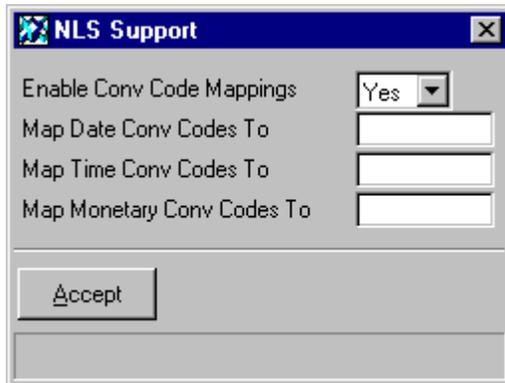
---

In PICK flavor accounts, if the code 'DI' is used in output conversions, UniVerse treats them as input conversions instead. Because all SB+ accounts use PICK style accounts, the code 'DI' should

not be used directly in your Basic subroutines. Instead, they should call `LOCALEINFO()` to obtain the correct conversion code for the user's current locale.

---

- For existing applications, you may choose to modify all the field definitions to use these NLS conversion codes, or you may leave the definitions unchanged and let SB+ map these conversion codes at run-time. This auto-mapping is controlled by the subscreen in SB+ **Control Parameters > F5-System Defaults > F8-NLS Support**. Enter **Yes** for the prompt **Enable Conv Code Mappings**, and enter the appropriate codes for date, time and monetary conversion. Normally these codes would be 'DI', 'TI' and 'MM' respectively. With these mappings enabled, all date, time and monetary fields will use the specified conversions at runtime. They overrides any conversion codes in both the field definitions and screen/report definitions.



- Finally, you may need to go through your application and check that it does not contain locale specific assumptions. It's also necessary to check how the data is stored to make sure it is locale neutral.

## PHANTOM Process Allowing Background Processing

The new PHANTOM process allows processes to be executed in the background. A temporary Job Schedule is created and executed as a background process (similar to Background Reports).

Processes executed in the background should not reply to Input/Output prompts.

### Syntax

PHANTOM [*processname*[,*param*][ (L)]

Normally the phantom process is executed immediately. However the L option forces a 'Launch' form to be displayed, allowing the default log and initiate details to be overwritten.

Executing from 'real' TCL/ECL:

MM PHANTOM [*processname*[,*param*][ (L)]

Executing from SB+ TCL Shell:

PHANTOM [*processname*[,*param*][ (L)]

Executing from a Prompt:

/PHANTOM

Executing from within a Paragraph:

DATA '*processname*[,*param*][ (L)'

EXEC 'PHANTOM'

Executing from within a Subroutine:

DATA '*processname*[,*param*][ (L)'

CALL SB.PROCESS('PHANTOM')

### Note

To execute a OE VERB use the '>:VERB' syntax, eg: >:WHO

## RedBack/SB+ Interface Demonstration

SB+ now contains a RedBack Application which demonstrates the RedBack/SB+ Interface.

### Location

The RedBack Application (back-end) exists in the SBDEMO account

The Web Interface (front-end) exists on the RedBack Gateway

### What is it?

It is an interface to SBDEMO.

#### Note



---

The back-end is written using RBOs (RedBack Business Objects).

The front-end is written in ASP (Active Server Pages).

---

### Functionality

It allows maintenance of SBDEMO files

- CUST
- BRANCH

It also allows a SB+ Process (RBO.TEST) to be executed, and displays version details of:

- SB+
- RedBack
- Operating Environment

### Requirements

The following are required in order to run the demonstration application:

### SB+ 5.0.3 or greater

- SBDEMO for SB+ 5.0.3 (or later)

### RedBack 3.5 or greater

- RedBack Application Server (back-end) installed on the same Server as SB+
- RedBack Gateway (includes SBDEMO front-end)

### Web Server with ASP support

## Setting up SB+/SBDEMO

1. Logto SBDEMO
2. RUN CHLIB ENABLE.RB  
Enter the RBDEFN path (RedBack back-end) if requested

## Setting up RedBack

The RedBack Application (SBDEMO) must be added to the RedBack .ini files:

rgwresp.ini (back-end)

rgw.ini (gateway)

### Program Example



```
rgwresp.ini
...
[sbdemo]
workdir=d:\sbplus\sbdemo
port=8261
...
[Accounts]
sbdemo
...

rgw.ini
...
```

```
sbdemo 123.123.123.123:8261
```

```
...
```

---

**Note**



Consult the RedBack Documentation for a more detailed explanation of the RedBack .ini files.

---

## Sources of Additional Information

For more information on this Application see the Readme in the Home menu (front-end)

For more information on RedBack, please contact your sales representative.

# Modifications

## Modified Function Key Behavior

From release 5 of SB+ Server, implicit <CR> or 'Accept' has been implemented after invoking processes linked to function keys or action bars. In previous releases of SB+ if the user entered a string followed by F5 for example (to invoke process assigned to Function Key 5) the content of the field was reset to it's original value. In other words, <Enter> or <Tab> key was required to accept the field. This was somewhat confusing for users in GUI mode, as it was not in line with standard Windows behavior.

In character mode, the function keys were processed only if the cursor was at the beginning of the input field. This has now been changed so the function key has an effect regardless of cursor positioning.

More detailed description of new functionality follows:

### Function Keys

Basically they behave the same way in both character and GUI modes. The functionality varies slightly depending on whether the control is returned to the input field after the function key is processed

**F1** - the current field content (text that has just been typed in) is preserved internally, help is displayed, field content is then restored but Common variable RECORD is never updated (because control is returned to the current input field).

**F2** - validation is run, if no errors encountered Common variable RECORD is updated with new value and then saved. If validation process generates an error, the control is returned to the input field and RECORD is not updated. This was also valid in the releases of SB+ prior to release 5.

**F3** - the current field content is preserved internally, intuitive help is displayed, field content is either restored or updated with selection. Common variable RECORD is never updated (because control is not returned to current input field).

**F4 - F10** (which also includes action bar options) - the validation is run, if no errors encountered Common variable RECORD is updated with new value and the associated action bar process is run.

## Button Events

These events are handled the same way as previously described for Function Keys.

## Menu Events

Applies to GUI mode only and is handled the same way as Function Keys.

## MPX, SBCom and ActiveX

Applies to GUI mode only.

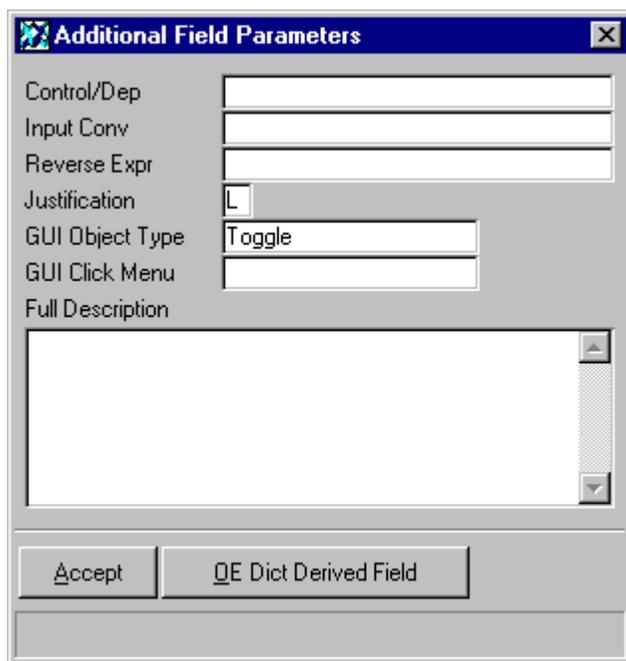
Current field content is set to the value of the STRING attribute of the current object after the Event is processed.

Validation is not run and RECORD is not updated, as control is not returned to the current input field.

## MV Grid Control Changes

The use of MV grid control introduced in release 4.4.3 of SB+ has been further expanded to allow 2 additional object types, combo option list and toggle. These objects present the user with a list of values from which to choose. The main difference is in how the choice list looks. The combo option list presents the user with list of values to choose from. For a toggle, there is a choice of only two values.

The values presented to the user are derived from data within the field or screen definition items, when the screen driver is generated. The object type to be used in MV grids can be specified via **F6 - Additional** in FD process or alternatively via **F5 > F6 - Additional Field Parameters** in Screen Definition tool.



Other MV grid features available in SB+ release 5 are column locking and re-arranging of the column order (re-arranging is only available in output forms). The order of the columns can be changed by clicking on column header, moving the column to the new position and then releasing the mouse button. In case of intuitive help (F3 lookups using ListView style), columns can be re-arranged only if there are no items left to be downloaded from the server (indicated by the icon in the bottom right-hand corner of the form).

Also, the sort direction is now indicated by the arrow in the column heading. For more details please refer to "Multivalued Grids" in the General Help Menu (**Technical Help > GUI Features**).

## Memo GUI Object Type

It is now possible to define Memo fields by setting the GUI object type on either the screen definition or the field definition to 'M'.

The functionality is identical to using GUI.TEXT.ED as the field's 'Process Before' process. Where you would previously specify an argument for GUI.TEXT.ED you can now specify a GUI object type of 'M' with the argument appended to it e.g. 'M2' is equivalent to using GUI.TEXT.ED,2 as the Process Before.

## Dialog Definitions Extended for GUI

Dialog definitions have been enhanced to allow specification of some new GUI only parameters.

A new option (F5-GUI Parms) has been added to the **Dialog Definitions** process (DIALOG.DEFN) allowing specification of the following parameters:

- Dialog Bitmap
- Option Bitmaps
- Checkbox Text
- De-emphasizing of options

Option Text	Bitmap Path	UnChecked	Checked
Cash		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Transfer		<input type="checkbox"/>	<input checked="" type="checkbox"/>
Credit		<input type="checkbox"/>	<input checked="" type="checkbox"/>
Charge		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## Dialog Bitmap

Allows specification of the bitmap to be displayed in the main dialog form.

## Option Bitmaps

It is also possible to specify bitmaps for individual dialog definition options (buttons).

## Checkbox Text

If specified a Checkbox will be displayed within the main dialog form. The text will be displayed to the right of the Checkbox.

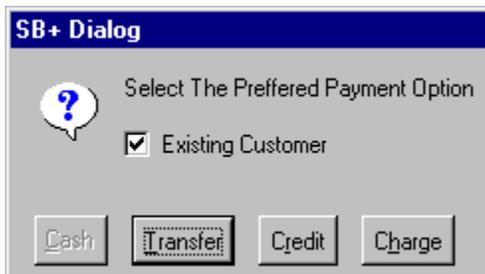
## Checkbox On

If set, the default Checkbox state is set to on. This is only available if Checkbox Text has been entered.

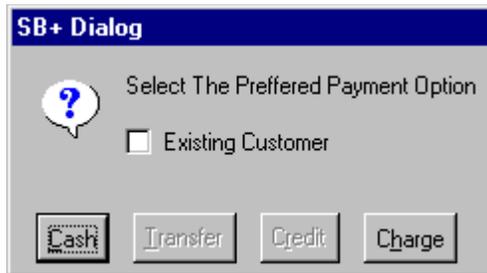
## De-emphasizing of Options

Options may be de-emphasized depending on whether the Checkbox is 'checked' or 'un-checked'. This is only available if Checkbox Text has been specified.

When executed the dialog is presented as shown below:



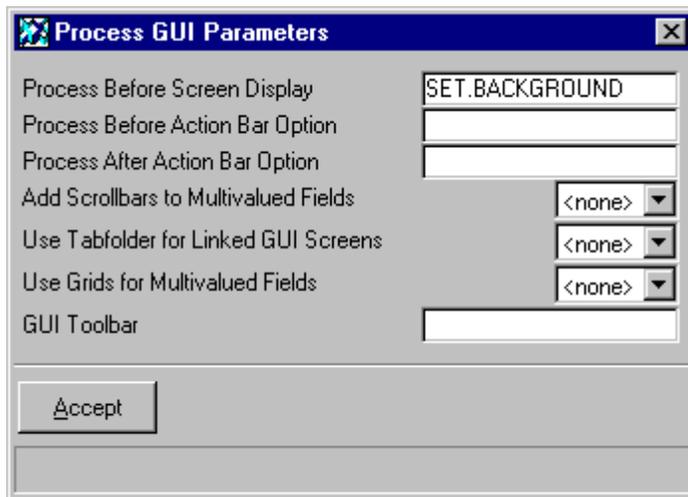
or if the checkbox is not checked then the dialog will appear as follows:



## Additional Process Before GUI Form Display

A new process hook has been provided for SB+ Input/Output processes that is only executed in GUI mode. The process is called after the form is created, but before it is made drawable. The purpose of it is to give developers a chance to manipulate GUI objects, like change background colors of the objects, without additional form painting.

Previously the only available process slot was 'Process After Display', but if any of visible attributes were changed the form display was not always neat.



## Other GUI Improvements

### SB+ GUI Desktop Settings Automatically Saved on Exit

The SB+ Desktop settings, i.e. size and position of the MainWin in GUI mode, are automatically saved on exit.

SB+ Explorer (SBX) parameters are also saved automatically on exit. These include:

- Size and position of the TreeWin form
- Process Id history
- Font
- Width of the Tree Pane
- ListView active/inactive

The Folder details (Branch text/headings and Icon, User Folder contents) are not saved automatically. To save the folder details, right-click on the Sysid node and choose Save Settings option.

For more details see "SBExplorer (SBX)" in the General Help Menu (**Technical Help > GUI Features**).

#### Note

---

The Save SB+ Desktop option has been removed from the System Menu of both the MainWin and TreeWin as it is no longer required.

---

### SB+ Explorer (SBX) Now Allows Specification of a Start Menu

The SBX process allows specification of the Start Menu as an optional parameter:

Example:

```
SBX, MAINMENU
```

The Start Menu will be used as the starting point for the Menu Structure branch and the standard rules will apply to the Tools branch. However, it will only be taken into account if the User does not have a start menu specified already via SB+ Security.

For more details see "SBExplorer (SBX)" in the General Help Menus (**Technical Help > GUI Features**).

## Progress Bar is now Available to Applications

The Progress Bar which is displayed within the Status Line may now be updated by applications via Paragraphs and User Subroutines.

Paragraphs will use the DISP command:

```
DISP 9,'message' AM position AM units
```

where:

position = position of progress bar in units

units = maximum number of units (if left blank 100 is assumed)

### Example

```
DISP 9, 'Downloading':@AM:100:@AM:1000 ;* 100/1000 or 10%
DISP 9, 'Processing XX.MASTER':@AM:50 ;* 50/100 or 50%
DISP 9, '' ;* Clear Progress Bar
```

User Subroutines will call SB.DISP():

```
CALL SB.DISP(9,'message' AM position AM units)
```

### Example

```
CALL SB.DISP(9, 'Uploading ':AM:1:AM:20) ;* 1/20 or 5%
CALL SB.DISP(9, '') ;* Clear Progress Bar
```

## Change to Separator Class (2 Pixels)

The width of separator object used in GUI forms has been modified from 3 to 2 pixels, to be in line with current design standards.

The MODIFY.GUI process, documented in the New Features section above, can be used to make similar changes to the separator dimensions in application forms if desired.

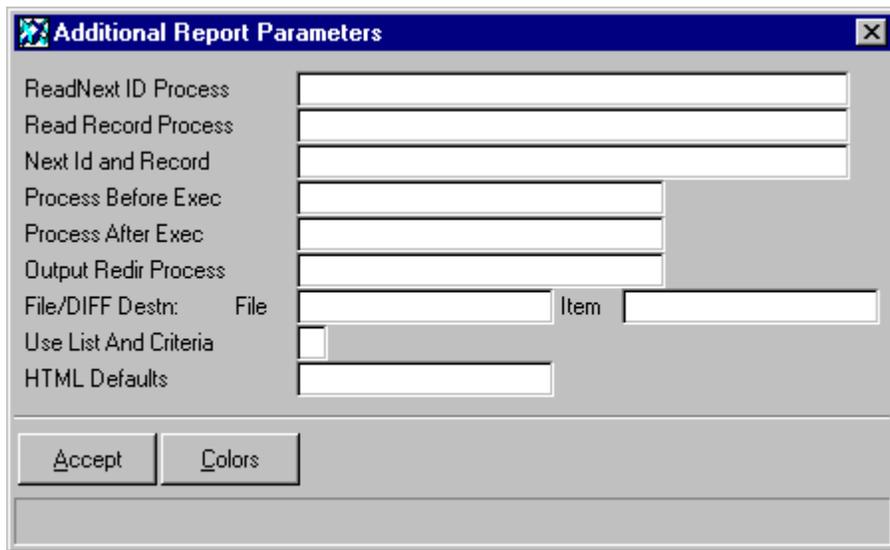
## **New process to reset GDS templates to 'factory defaults'**

A new process, CLEAN.GDS, has been added that will reset the standard GDS records back to their 'factory default' values. The 'factory defaults' are stored in DMCONT.

## Increased User Control of Report Writer Output Selection

In this release, user control over selection and processing of report output options has been extended by the following enhancements:

1. An extra field named **Output Redir Process** has been added to the **Report Parameters>More Report Parameters** (F6>F6) screen for each report. If a process is nominated in this field, it will override both the "Output Redir Process" field (previously named the "Prnt Redir Proc On" field in the module control record) and the normal output prompt issued by the report writer itself. If @RTN.FLAG is set when control returns from this routine then the report will be exited, otherwise the report writer will expect @VALUE to contain a valid output option letter for that report (eg. if the report does not support "background" processing but the redirection process returns a "B" in @VALUE, control will be passed back to the redirection process until either it returns a valid option letter for that report or @RTN.FLAG is set).



The screenshot shows a dialog box titled "Additional Report Parameters" with a close button in the top right corner. The dialog contains the following fields and controls:

- ReadNext ID Process: [Text Input]
- Read Record Process: [Text Input]
- Next Id and Record: [Text Input]
- Process Before Exec: [Text Input]
- Process After Exec: [Text Input]
- Output Redir Process: [Text Input]
- File/DIFF Destn: File: [Text Input] Item: [Text Input]
- Use List And Criteria:
- HTML Defaults: [Text Input]

At the bottom of the dialog, there are two buttons: "Accept" and "Colors".

2. SB+ now also supplies a standard redirection process (OUTPUT.SELECT) that can be used in this new field and takes as its parameter the (English) letters of the output options which are allowed. For example OUTPUT.SELECT,PX would cause the report writer to issue a prompt that only asks for, and only accepts, output to printer and auxiliary printer.

---

**Note** 

Although the option letters must be the standard English BDFPSX (Background, Dif, File, Printer, Screen, auXiliary), the prompt text itself still will be translated automatically, if a translation exists (field 145 in DMCONT record GENERAL.TEXT).

---

**Note** 

SB+ will insert the output options into the prompt string in exactly the same order in which those option letters were passed to OUTPUT.SELECT, giving the developer complete control over how the prompt appears to the user.

---

3. To enable the new HTML output style, it is necessary to include "H" in PARAM when OUTPUT.SELECT is called. However, there will not be a separate output option for HTML - HTML is an adjunct to the existing options and appears as a check box in GUI mode. If this box is checked and the option selected, the options with HTML capability will be enabled (Screen, auX printer or File) and the others disabled (Printer, Dif, Background).

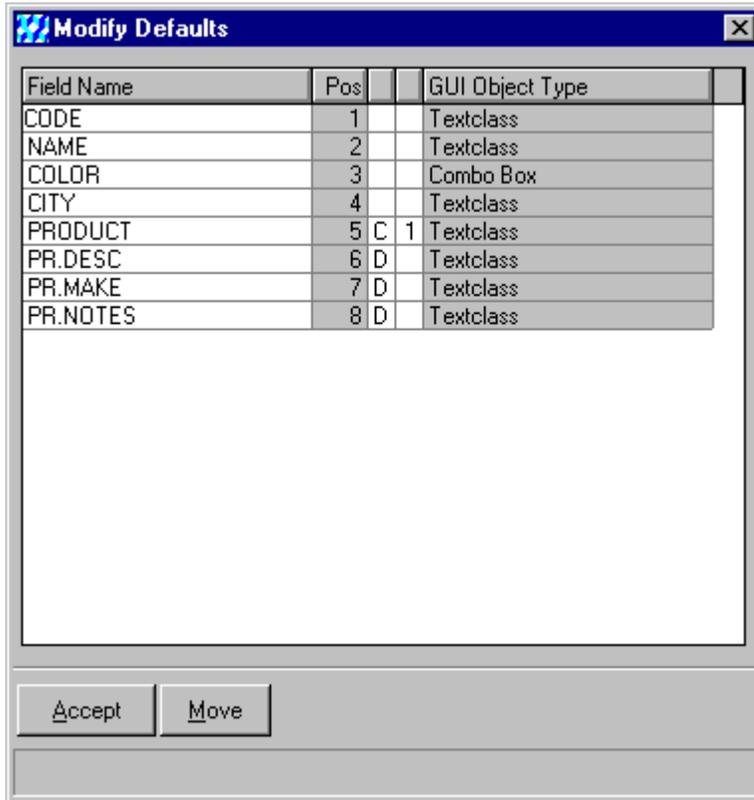
Users who wish to employ HTML output but are supplying their own redirection process, should return the appropriate option in VALUE preceded by 'H' (eg. 'HS' to display HTML output on the screen, 'HF' to write the HTML to a file, etc).

## Default Screen Generation Extended

Previously the **Generate Default Screen** function (F6>F5 in the screen definition tool) created one or more forms on the basis of one field per line. All multivalued fields were created as "controlling" with a depth of 1 and all prompts/labels were to the side. With the current release, this function has been expanded and refined by the addition of a separate input screen, similar to the existing **Resequenece Fields** screen, which allows for changes to be made after the fields have been selected but before the screen is actually built. At this point, it is now possible for the order of fields to be changed, multivalue controlling/dependency relationships to be altered and the depth of each multivalue set to be specified separately. As a result of this enhancement, SB+ will now initially make multivalue fields dependent wherever possible.

Each multivalue field will be created with its prompts/labels above the input field and as many dependent multivalue fields as will fit on the screen will be placed horizontally across the same line as the controlling field.

## Modify Defaults Screen



### Prompts

#### Field Name

The name of the field as entered in the initial **Generate Default Screen** screen.

#### Pos

Sequence number of each field in the form. (Display Only)

## **C/D**

C controlling,

D dependent.

Applies only to multi-value fields.

## **Window**

The number of rows which will represent the maximum number of lines of window space to be displayed on the screen at one time, if the field is defined as a multi-value. Window size does not indicate the maximum number of lines that may be entered, only the maximum to be displayed at any one time.

The extended dot (.) syntax, to force lines to be skipped between rows, is not available at this prompt.

## **GUI Object Type**

The object class for this field/object when the form is run in GUI mode. (Display Only)

## **Function Keys**

### **F2 - Accept**

Accepts the contents of the screen and initiates creation of the new form.

### **F8 - Move**

Allows the order of fields to be changed. This similar to the existing Resequene Fields option in the screen definition tool, except that in this case not only in the input order affected, but also the field position actually can be changed, as the screen has not been created yet.

## **Multi-lingual Support for Dialogs, Toolbars and Code Tables**

Multi-lingual support for Dialog Box Definitions is now available. In-line translations may be entered using the /OL process for the following attributes:

2 Text

3 Option Text

7 Help Reminder

The SB+ process /ML.MAP has been updated to display this information. A batch translation screen for Dialog Box Definitions is available via the /TRANSLATE process.

Multi-lingual support for Toolbar Definitions is now available.

In-line translation may be entered using the /OL process for the following attribute:

4 Help Text

The SB+ process /ML.MAP has been updated to display this information.

A batch translation screen for Toolbar Definitions is available via the /TRANSLATE process.

Multi-lingual support for Code Table Definitions is now available. In-line translations may be entered using the /OL process for the following attributes:

4 Display Text

5 Input Code

The SB+ process /ML.MAP has been updated to display this information.

A batch translation screen for Code Table Definitions is available via the /TRANSLATE process.

## **New Paragraph Statements for OLE & VBScript**

Support for OLE and VBScript was added to SBClient in release 4.5 of the product. To provide an alternative to DDE, some of this functionality has now been exposed in SB+ via the paragraph processor as follows:

Existing DDE paragraph statements:

```
DDECONNECT TO exp1 RETURNING handle [WITH PATH exp2 [MODE exp3]]
DDEREAD var FROM hExp USING exp1
DDEWRITE TO hExp USING exp1 SENDING exp2
DDEEXECUTE TO hExp SENDING exp1
DDEGETERROR FROM hExp
DDEDISCONNECT FROM hExp
```

Equivalent OLE paragraph statements:

```

OLEOPEN exp1 RETURNING handle WITH PATH exp2 [MODE exp3]
    either exp1 = "Word" or exp1 = "Excel"
OLEREAD var FROM hExp USING exp1
    for Word, exp1 = a bookmark
    for Excel, exp1 = a cell reference (eg. A3)
OLEWRITE TO hExp USING exp1 SENDING exp2
    for Word, exp1 = a bookmark
    for Excel, exp1 = a cell reference (eg. A3)
OLEEXECUTE TO hExp SENDING exp1
    exp1 = name of a macro
OLEGETERROR FROM hExp
OLECLOSE hExp
OLESAVE hExp [AS exp]

```

**Note** 

In the OLEOPEN statement, the "WITH PATH" clause is not optional (as it is for DDECONNECT) and currently OLE has been implemented only for Excel and Word. Also, an additional "OLESAVE... AS" statement, which has no equivalent in DDE, has been provided to make it easy to save a modified document or worksheet without having to close the session.

---

handle is a local or COMMON variable which will contain the handle returned.

hExp is used in other commands to refer to this connection. It must evaluate to the value in this variable. Normally it would refer directly to this variable.

VBScript paragraph statements:

```

VBSCRIPT CONNECT RETURNING handle
VBSCRIPT RUN CLIENT path_exp FUNCTION function ON hExp
    RETURNING result
VBSCRIPT RUN HOST oe_file_name_exp,oe_item_name_exp
    FUNCTION function ON hExp RETURNING result
VBSCRIPT RESET hExp

```

Both OLE and VBScript use the same timeout value as DDE (eg. the "DDE Timeout" field in the SB+ Control Parameters record).

The DDE, OLE and VBScript interfaces are only available when using SBClient. Additional information about the DDE, OLE and VBS capabilities of SBClient, together with the more general object handling routines provided in the SBClient host library, can be found in the SBClient documentation.

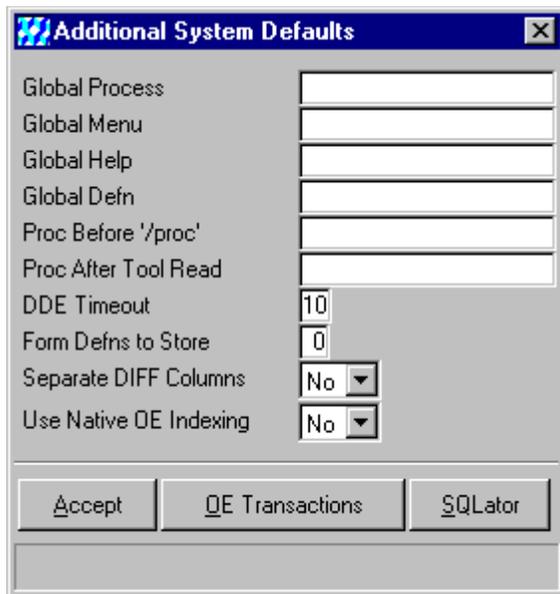
## Support for Alternate Indices in the Native OE

SB+ provides a standard and transportable implementation of Btrees for alternate key indices across all platforms. Support also exists on some platforms to use the native Btree facilities instead of those provided by SB+. Currently, use of native Btrees is supported on the following platforms:

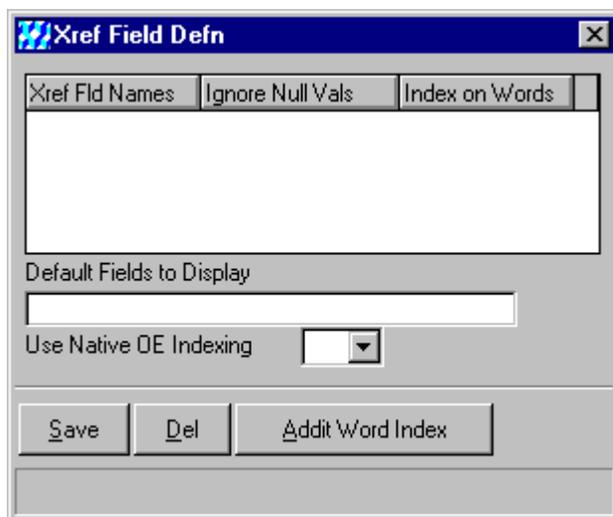
- D3
- UniData (from release 3.3.2)
- UniVerse (from release 8.3)

For these platforms, either native or SB+ Btrees can be used, depending on the setting of item <15,2> of the SB.CONTROL record in DMCONT. Normally this will be set only once, at installation time.

Please keep in mind that in case of Unidata and D3 this feature has been available in previous releases of SB+. In this release Universe has been added to the list of supported platforms and also the "Use Native OE Indexing" flag has been externalized, so that it is visible and can be maintained via SB.PARMS process (**SB+ Control Parameters>F5 System Defaults>F5 Additional System Defaults**).



In addition, it is now possible to override the global setting on a file by file basis, using the new "Use Native OE Indexing" field in the **Xref Field Defn** screen (**F10 Xref>Define X-ref Fields**, in the FD tool). This means that native indexing can be used for all files except for a selected few which use the SB+ Btree index files, or else SB+ Btree index files can be used for all files except for a selected few which use the native indexing.



## Improved SB+ Licensing Procedure

In order to simplify general licensing routine, the main SB+ licensing process has been modified to take out superseded values for number of SBClient and SBTerm users. The only remaining value related to number of users is Number of SB+ Users.

## Improved SB+ Upgrade Procedure

The SB+ Installation/Upgrade procedure has been enhanced in case of Unidata and Universe installations. Previously when the system was to be upgraded, number of manual steps were required, like renaming existing SB+ directories into .OLD, entering dummy SB+ licence and then running SA.TRANSFER.SBPLUS from ECL/TCL prompt.

In release 5 of SB+ the upgrade procedure has been fully automated in case of NT installations, and partially automated in case of Unix installations. When SB+ is initially loaded and install directory already contains SB+ directories, the dialog 'Do you want to preserve existing configuration' is presented to the user giving him the option to overwrite or preserve existing directories.

If the directories are to be preserved, they are then renamed into '.OLD' by the installation program. Upon logging in, previous configuration details (SB+ licence, user and group security, port configuration, etc.) are copied from older release into the new one.

**Note**



---

The upgrade procedure is fully automated for NT installations only, in case of Unix the existing SB+ directories still need to be renamed manually before new SB+ is loaded. But from that point no additional steps to perform the upgrade are required.

---

## Extended Tool Options

### SB+ Select Processes: New Option for Active SELECT Lists

For Intuitive Help (F3), the result of a selection is expected to be returned in VALUE. When the **S** option is specified in an SB+ selection process to create an active select list, the first value is automatically taken from the active list and moved into VALUE. To prevent this when the selection process is not being used as an F3, most users have simply specified the **D** option as well.

This is not altogether correct and has not always given users expected result. This Data stacking option is really intended for key fields (similarly, the **V** option is intended for multivalued/non-key fields). While the first key in the select list is not returned in VALUE if the **D** option is present (since a key field expects to get its value via an input, not from VALUE), other SB+ COMMON variables including RTN.FLAG may be set automatically to force input to return to the key field. This can upset processes which only check RTN.FLAG for error values and are ignoring its potential use for controlling the flow of processing.

A new option (**Y**) has now been provided. This acts similarly to **D** in that it does not move the first value selected in VALUE, but it differs from **D** by not setting other SB+ COMMON variables.

### Report Writer: New Break Option

The existing **D** option suppresses the printing of break details if only one record has been printed since the previous break. By default, the before and after processes associated with the break are also skipped. There is now a new break option **A** for "always run break processes". This causes the before and after processes still to be executed in such a situation.

This option has no affect unless the **D** option has also been specified.

### Report Writer: Break Fields now available in Headings & Footings

Break type fields ("Report Field Type" of **B**) may now be specified in the Header and Footer (**H** and **F**) sections of reports, in a similar way to the Break (**B**) section. At the "Derived Value" prompt, enter the name of a break field (as defined in the **Break Fields** prompt in the **Report Parameters** subscreen). This will cause the report writer to print the value of the field which prompted the last break.

## Report Writer: DIF Item Size Customization

The default behavior of Report Writer when output destination is 'Dif' (Data Interchange Format) is to split the generated output into number of smaller items. The maximum size of each item is by default 20K, or 500 lines, whichever condition is first met. When the DIF item reaches this limit (in size or number of lines), it is written into DIFF output file and next sequential item is initialized.

It is now possible to customize the size of generated items or even to use unlimited item size, in other words to create just one item that can be easily imported into MS Excel for example. To specify size other than default of 20K/500 lines, users have an option of creating the item SB.RD.DIFF in DMCONT file. Attribute 1 is expected to contain size in bytes and attribute 2 number of lines.

If the attribute 1 of SB.RD.DIFF item is set to 0, SB+ will create DIF item of unlimited size, i.e. it will create one final item containing the whole DIF output.

## SB+ Query: Increased User Control of Break Field Options

Previously, when an SB+ query was defined with one or more break fields, if no break options were specified for any of these fields then SB+ would automatically insert the options 'UVD' into the TCL statement it created. This was unsatisfactory as this process was not obvious to users. For release 5, the following changes have been made:

1. If the "Fields To Print" are built by using the F3 subscreen, then when a "Break" field is set to **Y** in the **Fields To Print** subscreen, SB+ will set the **Options** field to **UVD**, to show that this is what will be used by default. If the options are then set to null, SB+ will not insert any other options when the TCL statement is built.
2. If the **Fields To Print** F3 subscreen is not used, SB+ will insert the options it intend to use when the field is accepted, so that once again it will be obvious which options are going to be used.
3. If the query is changed to "totals only", then SB+ will strip out any **D** options in **Fields To Print**, otherwise fields which only generate one line of data would not be printed.
4. If the query is changed back from being "totals only" to being a full details report, then SB+ (re-)insert a **D** option in **Fields To Print**. This can then be overridden by the developer if desired.

## **SB+ Query: Extension to 'F' Output Option**

In release 4.5, the functionality of the SB+ Query tool was extended to allow the output destination to be to a file, provided the report was created with the Use SBSORT/SBLIST flag (in the **More Query Parameters** subscreen, reached via the F9-Addit option) set to **Y**. The SB+ Query tool has now been further extended to allow a file other than the default `xxWORK` file to be nominated.

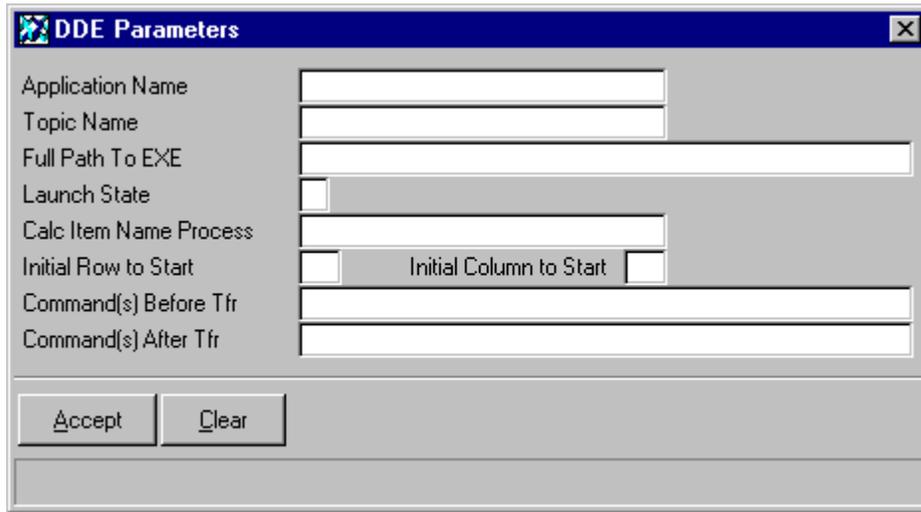
## **SB+ Query: Regeneration of Query Statement at Run Time**

When an SB+ Query definition is filed, SB+ automatically generates the OE query statement which will be used at run time and stores it in field 15 of the query definition record. This allows developers to edit and modify the statement if necessary, but means that there cannot be any substitution of SB+ COMMON variables at run time, since this could invalidate changes made by hand by a developer. There is now a new flag Regenerate at Run Time (in the **More Query Parameters** subscreen, reached via the F9-Addit option) which, if set to **Y**, causes SB+ to regenerate the OE query statement at run time, substituting the current values of any SB+ COMMON variables used in the selection criteria, heading or footing. If this flag is not set to **Y**, then SB+ will continue to use the existing OE query statement which was built when the Query definition was filed.

## **Pass To DOS Spreadsheet: Facility to Nominate Starting Cell**

In the **Pass To DOS Spreadsheet** tool, SB+ starts at the top left hand corner of the spreadsheet and fills the cells from there on according to the "pattern" of the data. The **Calc Item Name Process** allows users to override this default positioning. However, if a process name was specified in this field, then SB+ expected the nominated process to generate all the cell addresses for every item of data being sent to the spreadsheet - there was no separate option to nominate a starting cell but then have SB+ continue on its own from there.

For release 5, this has been rectified by the addition of two new fields: "Initial Row to Start" and "Initial Column to Start".



The screenshot shows a dialog box titled "DDE Parameters". It contains the following fields and controls:

- Application Name: text input field
- Topic Name: text input field
- Full Path To EXE: text input field
- Launch State: checkbox
- Calc Item Name Process: text input field
- Initial Row to Start: checkbox
- Initial Column to Start: checkbox
- Command(s) Before Tfr: text input field
- Command(s) After Tfr: text input field
- Accept: button
- Clear: button

---

**Note** ✓

These fields are incompatible with the "Calc Item Name Process" field. If that field contains an entry, then these fields are ignored as SB+ still expects the nominated process to generate all the cell addresses for every item of data being sent to the spreadsheet, including the initial cell.

---

## PD.LO & DOS.TDUMP: New 'Create Directory' Option

A new option C has now been provided for file downloads. If present and the destination directory does not exist, that directory will be created.

---

**Note** ✓

SB+ will only attempt to create one directory level - if more than one level does not exist, then the download will not proceed.

---

## **Screen Definition Tool Now Supports RW Graphics Definition Shortcut**

In the report definition tool, it is possible to completely specify boxes and lines at the first prompt after pressing F9 - for example, the response B40,6 can be entered to define a box 40 columns wide by 6 rows deep. In the screen definition tool, this was previously not possible. With this release, the screen definition tool now supports this same functionality.

## Security Improvements

### Changes to Type 5 Menu Security restrictions

In previous releases of SB+ when the menu structure was converted to type 5 the existing security needed to be re-entered for the new menu structure. For example, if SBMENU was converted to type 5 for GUI users, the security would have had to be entered for both MAINMENU and SBMENU\*MAINMENU items.

In release 5 of SB+ the changes have been made to avoid duplication, so if the security details are entered for type 1 menu, type 5 will use it too. Using the previous example, if TRANMENU is disabled for MAINMENU only, the restriction would apply to type 5 tree that contains TRANMENU as submenu. That way when a menu is converted to type 5, security is not lost and does not have to be re-entered.

Also, if there is specific security entered for a type 5 menu, the original menu will not be checked. In other words, it is still possible to have one set of security restrictions applied to type 5 menu and other to type 1.

### Extended Control Over Failed Login Attempts

SB+ login procedures have been changed to allow variation of the default SB+ maximum of three failed login attempts per connection and to provide the option of calling a user subroutine every time an incorrect password is entered at the SB+ login prompt. This subroutine has two arguments, Userid and Attempt #.

To activate this feature, create an item called VALIDATE.LOG in the DMSECURITY file. In attribute 1 of this item specify the maximum number of retries to be allowed (SB+ will continue to default to 3 if this field is left blank) and in attribute 2 specify the name of the user subroutine to call whenever an incorrect password is entered. The number of password retries applies system-wide, i.e. processes like User and Group security, Login and Keyboard timeout will also use it. The user subroutine needs to declare the two arguments, i.e. "SUBROUTINE USER.LOG(UID, ATTEMPT)" and must have an entry (catalog pointer) in the VOC.

There is also a control flag that needs to be set to 1 for new functionality to take the effect: (DMCONT) SB.CONTROL<31,5>. If this flag is not set the user subroutine will not be invoked.

## **Extended User Validation at Login**

SB+ login procedures also provide the option of calling a user subroutine after the correct password has been entered at the SB+ login prompt. This subroutine also has two arguments, Userid and Err. The Err flag has four return values:

```
ERR = 0  --> OK
ERR = 1  --> Error, Try again
ERR = 2  --> Error, fatal, log user off.
ERR = 3  --> Error, Try again but REDRAW SCREEN.
```

To activate this feature, create an item called `VALIDATE.USER` in the `DMSECURITY` file. In attribute 1 of this item specify the name of the user subroutine to call. The user subroutine needs to declare the two arguments, i.e. "SUBROUTINE ADDITIONAL.USER.CHECK(UID, ERRNO)" and must have an entry (catalog pointer) in the `VOC`.

## General Improvements

### SB+ Function Key Names Standardized

In the F6-Parms (**Report Parameters**) sub-screen of the Report Writer Definition (RD) tool:

"F8-More" has become "F6-Addit"

"F7-Convert Screen" has become "F8-Convert Screen"

"F6-Convert Query" has become "F7-Convert Query"

In the **Query Report Defns** screen:

"F9-Addit" has become "F6-Addit".

### SB+ Field Names Standardized

In the **Module Control Record** (HK.CONTROL), the **Prnt Redir Proc On** and **Prnt Redir Proc Off** fields have been renamed **Output Redir Proc On** and **Output Redir Proc Off** respectively.

Similarly, in the **Global Print Defaults** process, the **Global Prnt Redir On** and **Global Prnt Redir Off** fields have been renamed **Global Output Redir On** and **Global Output Redir Off**.

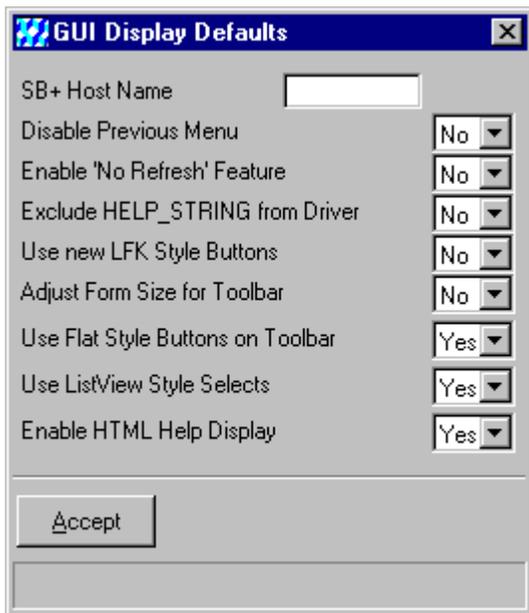
### Default Settings Changed in SB+ Control Record

Following settings are now turned on by default in a new installation of SB+:

- Use Flat Style Buttons on Toolbar
- Use ListView Style Selects

These settings are normally maintained via the SB.PARMS process:

**SB.PARMS > F8-Disp Defaults > F5-GUI Display Defaults**



**Internal Format of SB+ Messages Changed**

In release 5 of SB+ the format of messages (errors, warnings, dialogs) has been expanded to include reference to the SB+ function generating the messages. This has been done for support purposes only in order to be able to track the cause of the messages and provide quicker resolution to potential problems. Additional details related to the displayed message can be seen by pressing F3 key (the detail refer to the error code and content of COMMON variables MAINFILE and KEY).

**LISTU process expanded (UV/NT only)**

LISTU process has been expanded to display Account Name user is logged to. Also, column heading for the port number has been changed from 'User No' to 'Port'.

This change applies to SB+ for UniVerse/NT only.

## **ECLTYPE setting at module level (UDT only)**

Previously ECLTYPE setting was only available at user level (USER.FLAGS, F8-Shell Prefs). In this release the ECLTYPE can be set at module level (HK.CONTROL, **F5-Additional > F8-Shell Prefs**). This setting will have an effect only when the corresponding setting in the user flag is null (not set).





THE ART OF DATA MANAGEMENT

**Release Notes**

**DSB-5007**