

BC - SAP Style Guide



Release 4.6C



Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.

ORACLE® is a registered trademark of ORACLE Corporation.

INFORMIX®-OnLine for SAP and Informix® Dynamic Server™ are registered trademarks of Informix Software Incorporated.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.






HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA® is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Inhalt

BC - SAP Style Guide	15
About the SAP Style Guide	16
Software Ergonomics	18
What is Software Ergonomics?	19
The ISO Dialogue Principles	20
Existing Style Guides.....	21
Basic Goals and Design Guidelines	22
User Control	23
Reducing the Memory Load	24
Consistency.....	25
Software Ergonomics in the R/3 System	26
R/3 Standard Design Elements and How They Are Used.....	27
Introduction.....	28
Window Design Elements	29
Work Area Design Elements	31
The Different Levels in the R/3 System	33
Developing Ergonomic Software with the R/3 System.....	35
Object-Oriented Approach	36
Identifying Basic Objects and Actions	38
Outlining a Task: Hierarchy Structure and Work Processes.....	39
Structure Conversion: Menu Structure.....	40
Moving Around in the Structure: Navigation	42
Template Design in the Screen Painter	43
Details of the Interaction: Interaction Elements and Techniques.....	44
At last: Let's talk about the Menu Painter!	45
Important Additions to the Interaction: Help, Error and System Messages	47
Summary	48
General Design Guidelines.....	49
Prompts in Dialog Boxes.....	50
Categories	51
General Layout Guidelines.....	52
Confirmation Prompts After Returns	53
Confirming Actions	56
Confirming Actions After Previous Message.....	59
Choosing Between Two Related Alternatives With or Without Previous Message	60
Technical Notes	61
Cross-References	64
Modal Dialog Box	65
Guidelines	66
Pushbuttons or Function Keys in Dialog Boxes	68
Technical Notes	70
Cross-References	71
Levels in the SAP System	72
Main Menu Level.....	73
Application Level	74
Task Level	75

Cross-References	76
Accessing Tasks	77
General Procedure	78
Access via the Standard Menu	79
Access via Access Characters	80
Example	81
Cross-References	83
Exclusive OR Selections on Initial and Selection Screens	84
Two Alternatives Linked With	85
More Than Two Alternatives with Radio Buttons	87
Cross-Reference	88
Effects on Perception	89
The Effect of Proximity	90
The Effect of Similarity	91
The Effect of Closure	92
The Effect of Continuity	93
Cross-Reference	94
Selection Methods - Overview	95
Terms	96
General Guidelines	97
Types of Selection	99
Special Character '*' in the Selection Field	101
SAP Matrix	103
Technical Notes	104
Cross-References	105
Navigation Between Screens	106
The Screen Calling the Task	107
The Initial Screen	109
The Object Component List Screen	112
The Object Component Screen	117
The Sub-Component of Object Screen	125
Cross-References	128
Object-Oriented User Interface	129
General Explanations	130
Selecting Objects and Actions	132
Cross-References	133
Referencing Objects	134
Duplicating Procedure	135
Referencing Procedure	139
Cross-References	141
Searching in Structured Data	142
Wildcard Characters	143
Searching for Possible Entries (without Matchcode)	145
Searching for Possible Entries Using Matchcodes	147
Processing a Set of Objects	149
Further Uses of	150
Cross-References	151
Title Bar	152
General Guidelines	154
Guidelines for the Various Levels	155
Naming General SAP Screens	157

Title Bar in Dialog Boxes	161
General Notes	164
Technical Notes	165
Cross-References	166
Control Elements	167
Control Elements - Overview	168
Types of Control Elements	169
Using Control Elements	173
Cross-References	175
Checkboxes (Control Element)	176
Use	177
Design Guidelines	178
Technical Notes	182
Cross-References	183
Access Characters	184
Use	185
Access Character Overview	186
Alphabetical List	187
Classified List	189
Technical Notes	191
Cross-References	192
Radio Buttons (Control Element)	193
Overview	194
Design Guidelines	195
Technical Notes	199
Cross-References	200
Command Field	201
Scroll Bars (Control Element)	202
Pushbuttons With No Fixed Position (Control Element)	204
Use, Appearance and Control	205
Arrangement on Data Screens	207
Arrangement on Initial Screens	212
Pushbuttons with Icons on Data Screens	215
Pushbuttons with Icons on Initial Screens (Tools)	220
Report Selection Screens	222
Pushbuttons in Dialog Boxes	223
Technical Notes	225
Cross-References	226
Application Toolbar (Control Element)	227
Use and Position	228
Design Guidelines	229
Icons in the Application Toolbar	231
Icons on Pushbuttons in Dialog Boxes	234
Technical Notes	236
Cross-References	237
Input/Output Fields (Control Elements)	238
Purpose	240
Appearance	241
Design Guidelines	243
Scrollable Input/Output Fields	247
Technical Notes	249

Overview of the Field Attributes	250
Cross-References	252
Fields - Positioning	253
Arranging Fields According to Screen Sequence	255
Grouping Several Fields in Blocks	257
Gathering Several Blocks to Form a Block Group	259
Arranging Several Block Groups on a Screen	261
Cross-References	266
Function Keys - Overview	267
Illustration and Purpose	268
Guidelines - Current Conventions	269
Overview - Current Conventions	271
Guidelines - New Conventions	273
Overview - New Conventions	274
Technical Notes	276
Cross-References	277
Function Key Menu	278
Group Box (Control Element)	280
Purpose and Appearance	281
Design Guidelines: Basic Rules	282
Design Guidelines: Arrangement	283
Design Guidelines: Screen Examples	285
Technical Notes	288
Cross-References	289
Long Screens	290
Use	291
Screen Design	292
Checks	293
Cross-References	294
Menus (Control Elements)	295
Purpose	296
Menus in the R/3 System	297
Shortcuts	298
Cross-References	299
Check Text (Control Element)	300
Status Display (Control Element)	302
Status Bar	305
Standard Toolbar	307
Purpose and Appearance	308
Design Guidelines	310
Icons for the Standard Toolbar	311
Technical Notes	312
Cross-References	313
Tab strips	314
Using a tab strip	315
Tab strip components	317
Programming tab strips	319
Defining a tab strip in the Screen Painter	321
Field Names (Control Element)	324
Use and Appearance	325
Design Guidelines	326

Icons as Field Names.....	328
Technical Notes	329
Cross-References	330
Possible Entries Pushbutton (Control Element)	331
Use	332
Effect on the Screen design	335
Technical Notes	337
Cross-References	338
Tables and Lists	339
Table Control	340
Functionality	341
Description of the Options	343
Use and Effect of the Options	345
Appearance	350
Technical Notes	353
Cross-References	354
Entry and Display Tables.....	355
Field Attributes	356
Page Display	359
Search for Lines	360
Cross-References	361
Selection Tables and Lists	362
Multiple Selection Lists.....	363
Single Selection Lists	365
Matrices.....	366
Cross-References	367
Arranging Table Fields	368
Alignment of Output Fields.....	369
Arranging Column Titles.....	370
General Design of One-Line Tables	371
Multiple-Line Tables	373
Technical Notes	374
Cross-References	375
Tables - Scrolling	376
Scroll Functions and Scroll Keys for Vertical Scrolling	377
Horizontal Scrolling	379
Scroll Bars	380
Technical Notes	381
Cross-References	382
Tables - Maintenance	383
Functions for Inserting and Appending Lines.....	384
Copy, Move, Delete, and Sort Functions	387
Working With the Clipboard	392
Cross-References	393
Lists - General Issues	394
Data Structures and Forms of Presentation.....	395
Classifying Lists	397
Cross-References	400
Lists - Colors and Lines.....	401
General Information on List Design	402
Color Scheme	403

Header Information	405
One-Line Lists	407
Highlighting Columns, Lines, and Cells	411
Multiple-Line Lists	414
Hierarchical-Sequential Lists.....	416
Case 1: Hierarchy Information With Few Criteria and Key Figures	417
Case 2: Hierarchy Information With Many Criteria and Key Figures	419
Matrix Lists	421
Object Lists.....	423
Hierarchies	424
Technical Notes	426
Cross-References	428
Lists - Hotspots	429
Lists - Symbols and Icons	432
General Remarks on Symbols and Icons in Lists	433
Symbols for Displaying Hierarchies	436
Symbols for Reporting.....	438
Symbols for Office Applications (Mail)	439
Symbols for Displaying Status	441
Symbols for Indicating the Status of Objects and Values	443
Symbols for Displaying Ratings	444
Navigation Symbols	446
Technical Notes	447
Cross-Reference	449
Functions - General Guidelines and Overview.....	450
Functions - Overview	451
Overview List.....	452
Cross-References	455
Defining Functions	456
Technical Notes	457
Cross-References	459
Navigation Functions - Overview	460
Overview of Navigation Options.....	461
Comparison of Exit, Back and Cancel	464
Cross-References	466
Functions - List.....	467
Overview	469
Cancel (Function)	470
Select All, Deselect All (Functions)	473
Other <object> (Function)	474
Other <object component> (Function).....	476
Change (Function).....	477
Create (Function).....	479
Display (Function)	481
Display <-> Change (Function)	483
Execute (Function)	485
Cut (Function).....	486
Choose (Function).....	487
Edit, Single Processing, Collective Processing (Functions)	489

Exit or End <application> (Function)	492
Select Block (Function)	494
Print (Function).....	496
Editor or Long Text (Function).....	497
Insert or Insert <object> (Function).....	498
Possible Entries (Function).....	499
Paste (Function)	502
Enter, Single Entry, Collective Entry (Functions)	503
First Page, Last Page (Function)	505
First Screen, Last Screen (Function)	507
Generate (Function)	508
Help (Function)	509
Open (Function).....	512
Copy (Function)	513
List (Function)	514
Delete or Delete <object> (Function).....	515
Select/Deselect (Function)	516
Menu Bar (Function)	518
Hold (Function).....	519
New Lines (Function)	520
Check (Function)	521
Reset (Function)	522
Close (Function)	523
Save or Post (Function)	524
Save Without Check (Function)	526
Sort (Function).....	527
Cancel/Reverse (Function)	529
Total (Function)	530
<Object Component> up/down/left/right (Functions)	531
Move (Function).....	532
Insert Many (Function)	534
Previous Page, Next Page (Functions)	535
Previous Screen, Next Screen (Functions).....	536
Previous <object component>, Next <object component> (Functions)	538
Continue (Function)	540
Undo (Function).....	543
Back (Function)	544
Menus - General Guidelines and Overview	547
Menus - General Structure	548
General Design Guidelines	549
Structure and Access	552
Technical Notes	553
Cross-References	554
Menu Bar at Application Level	555
Design Guidelines	556
Technical Notes	557

Cross-References	558
Menu Bar at Task Level	559
Design Guidelines	560
Cross-References	562
Menus	563
<Application> (Menu).....	564
Edit (Menu).....	565
Pull-Down Options	567
Cross-References	569
Help (Menu).....	570
Pull-Down Options	571
Technical Notes	572
Cross-References	573
<Object> (Menu)	574
Illustration	575
Pull-Down Options	577
Cross-References	579
Goto (Menu)	580
Environment (Menu).....	583
Extras (Menu).....	585
Texts	587
Using Texts on Screens.....	588
Abbreviations.....	593
Guidelines	594
Examples for the ABAP Dictionary.....	597
Technical Notes	599
Cross-References.....	600
Formulations.....	601
Highlighting.....	603
General Messages	604
Guidelines	605
Examples of Good and Bad Error Messages	607
Technical Notes	609
Cross-References.....	610
Message Types and Error Groups.....	611
Message Types	612
Error Groups	614
Technical Notes	617
Cross-References.....	618
Upper/Lowercase	619
Hyphenation.....	620
Translation	621
User Interface.....	622
Screen Painter	624
ABAP Text Pool	625
Further Translation Problems	627

Cross-References	629
International Terminology Lists	630
International Terminology Lists - Overview	631
Standard Function Names	633
Menu Bar Options	635
Documentation not Available in Release 4.6B	636
Documentation not Available in Release 4.6B	637
Documentation not Available in Release 4.6B	638
Documentation not Available in Release 4.6B	639
Documentation not Available in Release 4.6B	640
Terms of the User Interface	641
Overview of Standard Functions	646
Functions	647
Legend	656
R/3 Icons and Symbols	657
Icons	658
Icons on Pushbuttons in the Standard Toolbar	659
Icons on Pushbuttons in the Application Toolbar	660
Pushbuttons with Icons on Data Screens	662
Pushbuttons with Icons on Initial Screens (Tools)	664
Application-specific Functions (ABAP/4 Workbench)	665
Special Icons for Networks	666
Report Selection Screens	667
Specific Icons on Pushbuttons in Dialog Boxes	668
Icons for Messages and System Prompts	669
Icons as Field Names	670
Icons as Status Displays	671
Symbols	672
Symbols for Reporting Tasks	673
Symbols for Office Applications	674
Symbols for Use as Status Displays	675
Symbols showing the Status of Objects, Numbers, Data Columns etc.	676
Symbols for Navigation Functions	677
Overview of the Symbol Character Font SAPDings	678
Overview of Menu Bars	679
General Remarks on Menu Bars	680
Menu Bar at Task Level	681
<Object>	682
Edit	683
Other Menus	684
Graphical Overview of SAP GUI Terms (German/English)	685
Glossary	687
Prompts	690
Abbreviations	691
Pull-down menu	692
Modeless dialog box	693
Checkbox	694
Task	695
Task level	696

Task function	697
Work area	698
Application level	699
Application area menu	700
Access character	701
Radio button	702
Command field	703
Command list	704
Screen	705
Screen type	706
Scroll bar	707
CUA	708
Dialog box	709
Fast path	710
Pushbutton	711
Pushbutton with a fixed position	712
Pushbutton with no fixed position	713
Application toolbar	714
Levels	715
Input/output fields	716
Single-selection list	717
Initial screen	718
ENTER pushbutton	719
EU 90/270/EWG	720
Error message	721
Field name	722
Fields	723
Window	724
Function	725
Function name	726
Function key	727
Function key menu	728
Group box	729
Group heading	730
Highlighting	731
ISO 9241-10 (Dialogue principles)	732
Cascading menu	733
Macintosh	734
Multiple-selection list	735
Message	736
Message type	737
Menu	738
Menu bar	739
Menu Painter	740
Modal dialog box	741

Session.....	742
Motif	743
Required-entry fields	744
Navigation	745
Standards and guidelines (national, international).....	746
Primary window	747
Main menu level.....	748
SAP matrix	749
Screen Painter	750
Secondary window	751
Confirmation prompts.....	752
Status bar	753
Standard toolbar.....	754
Table	755
Title	756
Title bar.....	757
Hyphenation	758
Possible entries pushbutton	759
Windows.....	760
Clipboard.....	761

BC - SAP Style Guide

About the SAP Style Guide

About the SAP Style Guide

About this Project

The SAP Style Guide provides guidelines for developing the user interface for R/3 applications. The aim of the Style Guide is to support developers when creating user-friendly and standard software for the SAP System. The Style Guide concerns all the applicable SAP design guidelines. The SAP Style Guide is based on platform-specific style guides such as the IBM CUA Guidelines, the Windows Style Guide or the Apple Macintosh Human Interface Guidelines, which have been adapted and added to meet SAP's business needs.

Applicability of the Style Guide

This version of the Style Guide includes the standards 1-24 as well as all its appendices and updates.

New technical features and innovations in the field of ergonomics result in new or changed design guidelines. Therefore, the Style Guide constantly has to be adapted. Until these additions and changes are included in the Style Guide, you have to use the Style Guide together with the corresponding new standards.

Structure of the Style Guide

The first two chapters of the Style Guide describe the basic requirements for developing ergonomic software and how they are implemented in the R/3 System. The following chapter contains general guidelines for the SAPGUI design. The remaining chapters deal with the control elements used in SAPGUI, tables and lists, functions, menus and formulation and arrangement of texts.

Structure of the Keywords

The individual chapters list keywords which are organized as follows: each keyword starts with a short explanation of the topic dealt with, followed by a graphical illustration and relevant guidelines. If possible, examples and explanations are given for every guideline. Technical notes are provided to simplify the use of these guidelines during the development of tasks. If a keyword relates to one or more of other keywords, appropriate cross-references are given.

Overviews

A summary of all standard functions with their pull-down menus, function keys and pushbuttons as well as their syntaxes is given in [Overview of Standard Functions \[Seite 646\]](#). [Overview of Menu Bars \[Seite 679\]](#) provides an overview of the menu bars at the individual levels of the SAP System, in particular the menu bar at task level. [Graphical Overview of SAP GUI Terms \(German/English\) \[Seite 685\]](#) gives a graphical overview of the most important R/3 user interface terms in both German and English. For definitions of the most important R/3 user interface and ergonomics terms, please refer to the [Glossary \[Seite 687\]](#).

Software Ergonomics

[What is Software Ergonomics? \[Seite 19\]](#)

[The ISO Dialogue Principles \[Seite 20\]](#)

[Existing Style Guides \[Seite 21\]](#)

[Basic Goals and Design Guidelines \[Seite 22\]](#)

[User Control \[Seite 23\]](#)

[Reducing the Memory Load \[Seite 24\]](#)

[Consistency \[Seite 25\]](#)

What is Software Ergonomics?

Software design is not just a matter of finding an algorithm for the purpose of accomplishing a given task. With software systems becoming more and more complex, the appearance of the system to the user is becoming ever more important. The design of user interfaces following ergonomic principles facilitates the learning process and minimizes the need for training when new applications are introduced into the workplace, resulting in increased efficiency.

This user-centered approach to the development of interactive systems is software ergonomics.

What are "ergonomic principles"? And in how far do they increase efficiency? A user interface should allow the user to interact with the system simply and naturally. The system should facilitate the user's tasks rather than constituting an obstacle in itself. The following sections explain how these goals can be accomplished.

Selected Bibliography

For further information on software ergonomics, please refer to the following books:

- Eberleh, E., Oberquelle, H. & Oppermann, R. (Hrsg.) (1994). *Einführung in die Softwareergonomie*. Berlin, New York: de Gruyter.
- Fowler, S. L. & Stanwick, V. R. (1995). *The GUI style guide*. Boston: AP Professional.
- Helander, M. (Ed.) (1988). *Handbook of Human-Computer Interaction*. Amsterdam: North-Holland.
- Shneiderman, B. (1987). *Designing the user interface*. Reading MA: Addison-Wesley.
- Wandmacher, J. (1993). *Softwareergonomie*. Berlin: de Gruyter.
- Zeidler, A. & Zellner, R. (1994). *Software-Ergonomie*. München: Oldenbourg.

These books also contain extensive ergonomics bibliographies.

The ISO Dialogue Principles

The ISO Dialogue Principles

The dialog principles identified as being important for the design and evaluation of a interactive software system are defined in the standard ISO9241-10. The ergonomic requirements described in the following section are also documented in one form or the other in the books mentioned in the section Existing Style Guides. The European Community Directive 90/270 also lists similar recommendations.

Suitability for the Task

A dialog supports suitability for the task, if it supports the user in the effective and efficient completion of the task. The dialog presents the user only those concepts which are related to the user's task.

Self-Descriptiveness

A dialog supports self-descriptiveness, if each dialog step is immediately comprehensible through feedback from the system or is explained to the user on his or her requesting the relevant information.

Controllability

A dialog supports controllability, if the user is able to maintain direction over the whole course of the interaction until the point at which the goal has been met.

Conformity with User Expectations

A dialog supports conformity with user expectations, if it corresponds to the user's task knowledge, education, experience, and to commonly held conventions.

Error Tolerance

A dialog supports error tolerance if, despite evident errors in input, the intended results may be achieved with either no or minimal corrective action having to be taken. Errors should be explained to the user for him or her to correct them.

Suitability for Individualization

A dialog supports suitability for individualization, if the dialog system is constructed to allow for modification to the user's individual needs and skills for a given task.

Suitability for Learning

A dialog supports suitability for learning, if it guides the user through the learning stages minimizing the learning time.

These requirements are not mere theoretical aspects, they will be implemented as laws in future. This means that user will be able to make legal claims against companies whose software systems do not comply with above mentioned ergonomic requirements.

Existing Style Guides

To put these general requirements in more concrete terms, ergonomic interface design principles have been, and still are being developed on different levels of abstraction. We cannot go into all of them, but more information can be found in:

- Apple Computer, Inc. (1992). *Macintosh Human Interface Guidelines: The Apple Desktop Interface*. Reading, MA: Addison-Wesley.
- IBM (1991). *SAA/CUA Advanced Interface Design Guide*. Report SC34-4289-00.
- Microsoft Corporation (1992). *The Windows Interface: An Application Design Guide*. Microsoft Press.
- Open Software Foundation (1990). *OSF/Motif Style Guide*. Rev. 1.2. Englewood Cliffs, New Jersey: Prentice Hall.
- Sun Microsystems, Inc. (1990). *OPEN LOOK Graphical User Interface Application Style Guidelines*. Reading, MA: Addison-Wesley.

Basic Goals and Design Guidelines

Basic Goals and Design Guidelines

The following section deals with the goals and the resulting user interface design guidelines as described in IBM's *CUA Style Guide*.

The *CUA Style Guide* mentions the following **goals** for designing successful user interfaces:

- higher user productivity
- greater user satisfaction with the product
- lower user error rate

The following **design guidelines** for interface designers result from these goals:

1. Give the user control
[User Control \[Seite 23\]](#)
2. Reduce the user's memory load
[Reducing the Memory Load \[Seite 24\]](#)
3. Keep interfaces consistent
[Consistency \[Seite 25\]](#)

User Control

The user is in control when a system is flexible, that is, it allows the user to select from among the methods of accessing a function. Applying the following guidelines increases the user's sense of control:

Cautious use of different modes

A mode is a status of a product which restricts the interactions available to the user at any given time, for example, a line draw mode or a write mode. Modes also affect the behaviour of the application. Modes can be used to guide the interaction of the user with the application. The user should always be in control of the application, not vice versa. In the different modes, the same action can produce different results within an application.

If modes must be used, they should be visually obvious (for example, the pointer can change shape), easy to learn, and easy to get out of. While one particular mode is active, the user should be able to continue work elsewhere in the product.

Descriptive, helpful messages

If the product cannot interpret a user action, the user should be informed. Messages should state the problem or error clearly and offer possible solutions.

Rapid response and undo functions

The user should receive immediate and tangible feedback for actions within an application. If this is not possible for the time being, the user should receive a corresponding message. An undo function should be implemented to allow the user to reverse previous actions. If this is not possible, the user should be informed and allowed to choose another action.

Varying skill levels

Inexperienced users should be able to use the application without consulting manuals etc. and to improve their performance during its use (learning by doing), while advanced functionality such as hidden mechanisms and shortcuts should be available for experienced or expert users. In general, effective interface design starts with taking the various levels of user knowledge into account.

Transparency of the user interface

Users want to accomplish tasks, not to use computers or particular tools, so the user interface should be easy to understand and not demand more of the user's attention than the job at hand.

Modification of the interface to the user's individual preferences

Because users' abilities and experiences vary, they should be able to customize aspects of the interface such as screen colors and color intensity, the structure and content of menus, the sequence of the steps in a given task, and so on.

Reducing the Memory Load

Reducing the Memory Load

The user should not be required to remember information that can be provided by the system. Applications can facilitate users' tasks by providing helpful information while the user is working on a given task. Designers should also provide default values, and values selected should be saved.

Define meaningful, unique object classes

The designer should look at a task from the user's perspective and ensure that the attributes of the objects defined actually support the user in performing the given task.

The designer should clearly define the attributes of each object and should establish a hierarchy of object classes based on these definitions. The number of classes should be kept to a minimum and each class should be unique.

Before implementing a new object class, designers should try to modify an existing class for new objects.

Make objects recognizable and concrete

If the system uses objects which correspond to objects in the real world, this allows the user to transfer knowledge of these objects from real-world experiences. This facilitates the learning process.

Screen objects need to have a distinct appearance that the user can easily recognize and quickly understand. The appearance of the objects should be similar to real-world analogues.

Consistency

Consistency helps the user transfer familiar skills to new situations. The user can apply the knowledge learned from one application to another application. To create consistent user interfaces, designers should develop paradigms for the implementation of general functions throughout an application.

Consistency within a task

To convey a feeling of continuity to the user, a product should provide information which enables the user to relate cause and effect.

In addition, a user should be able to complete one or more related tasks without having to change between different types of input.

Users need to be able to anticipate the natural progression of each task. The same action should always have the same result. The system should provide functions which are relevant for the given task.

Consistency within applications and between different application versions

A new application or a new version of an application should broaden the knowledge of the user. Changing the behavior of an object from one version to the next might confuse the user. To satisfy both new and experienced users, offer both the old and the new behavior of an object and allow the users to choose their personal preference.

Aesthetic appeal

By skillful use of colors, proximity, overlapping, different forms and other types of visual communication, an interface can be designed to increase the user's efficiency, effectiveness, and satisfaction.

Using visual metaphors

A good graphical representation allows the user to find a relationship to his or her existing knowledge. Effective metaphors can extend the user's conceptual model.

The **SAP Style Guide** is to support developers when creating a consistent user interface of the R/3 System.

Software Ergonomics in the R/3 System

[R/3 Standard Design Elements and How They Are Used \[Seite 27\]](#)

[Introduction \[Seite 28\]](#)

[Window Design Elements \[Seite 29\]](#)

[Work Area Design Elements \[Seite 31\]](#)

[The Different Levels in the R/3 System \[Seite 33\]](#)

[Developing Ergonomic Software with the R/3 System \[Seite 35\]](#)

[Object-Oriented Approach \[Seite 36\]](#)

[Identifying Basic Objects and Actions \[Seite 38\]](#)

[Outlining a Task: Hierarchy Structure and Work Processes \[Seite 39\]](#)

[Structure Conversion: Menu Structure \[Seite 40\]](#)

[Moving Around in the Structure: Navigation \[Seite 42\]](#)

[Template Design in the Screen Painter \[Seite 43\]](#)

[Details of the Interaction: Interaction Elements and Techniques \[Seite 44\]](#)

[At last: Let's talk about the Menu Painter! \[Seite 45\]](#)

[Important Additions to the Interaction: Help, Error and System Messages \[Seite 47\]](#)

[Summary \[Seite 48\]](#)

R/3 Standard Design Elements and How They Are Used

[Introduction \[Seite 28\]](#)

[Window Design Elements \[Seite 29\]](#)

[Work Area Design Elements \[Seite 31\]](#)

[The Different Levels in the R/3 System \[Seite 33\]](#)

Introduction

What is a User Interface?

A user interface comprises all the technical attributes and functions which enable users to exchange information with a computer system. The way information is arranged on the screen and the user can interact with an application by using either a keyboard or a mouse is one of the characteristics of the user interface. The user interface of the SAP R/3 System (SAP GUI = SAP Graphical User Interface) is based on the principles of graphical user interfaces and follows an object-oriented approach.

[Object-Oriented User Interfaces \[Seite 129\]](#) allow designers to develop an integrated work environment which contains elements, called objects, with specific attributes. The user can use these elements to carry out particular functions. An object is a visible part of the user interface. To accomplish a particular task, the user can work with an object independently of other objects. A typical object in the SAP System is a document. Typical actions are to create, display or edit a document.

Style Guides

A graphical user interface designed in accordance with the object-oriented approach is characterized by particular interface elements and particular behaviour. The guidelines laid down in the style guides of various manufacturers and committees are of a general nature because they like to see their guidelines widely used. They also want to cover a large number of different applications.

The SAP Style Guide

To achieve a largely standardized user interface and system behavior, the SAP Ergonomics Group has adopted and supplemented these general guidelines to meet the requirements of SAP's integrated R/3 business applications. This adjustment process resulted in company-specific standards for the R/3 user interface laid down in the SAP Style Guide. The guide aims to provide SAP software developers with an overview of, and information about the SAP GUI.

SAP System Presentation

The SAPGUI runs on certain computer frontend systems which have the task of displaying the R/3 user interface. The R/3 System currently supports the following platforms: MS Windows, OSF/Motif, Presentation Manager (OS/2) and Apple Macintosh. Depending on the platform used, the SAP GUI displays may differ slightly, usually only in ways not covered by these design guidelines. If some of these differences do affect these guidelines, particular reference is made.

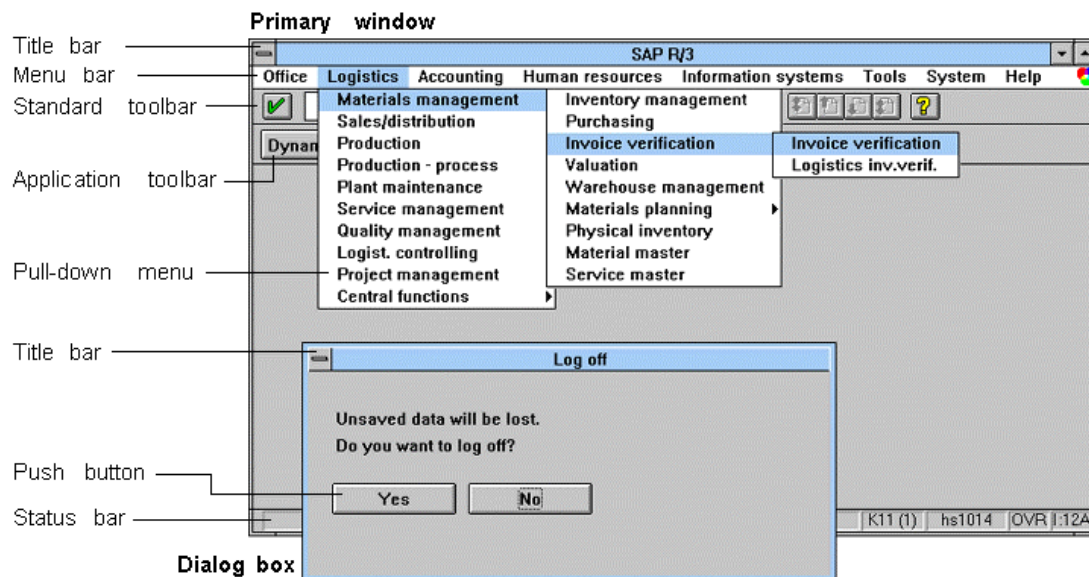
Since MS Windows is so widely used, all R/3 screen dumps shown in this project have been made under MS Windows.

Window Design Elements

[Drucktastenleiste \[Seite 204\]](#)

This section provides an overview of the R/3 interface elements and their behaviour.

R/3 Design Element Names



Window Types

In the R/3 System, we distinguish between primary windows and secondary windows. In contrast to the primary window, the secondary window is a subordinate window that appears on the current window when the user activates a particular action in a primary or secondary window. It disappears from the screen when the user performs a task-related action. Closing the primary window, however, terminates the current task altogether.

In SAP terms, secondary windows are referred to as "dialog boxes". We distinguish between modal and modeless dialog boxes:

- A [Modal Dialog Box \[Seite 65\]](#) is a dialog box which the user has to close before he or she can return to the higher-level window.
- With modeless dialog boxes, the user can continue to work on higher-level windows without being required to close the dialog box first.

There are so far no modeless dialog boxes in the R/3 System (modeless windows are to follow in one of the future releases).

Window Design Elements

Window Areas

SAP windows have the following structure:

- **Title bar:**

The text displayed in the [Title Bar \[Seite 152\]](#) is to inform the user about the current position and activity in the system.

- **Menu bar and pull-down menus (only available in primary windows):**

The menu bar looks different in the various [Levels in the SAP System \[Seite 72\]](#). While the Main Menu Level contains all applications in the menu bar and the menu of the Application Level contains all tasks of the application in question, the various screens at Task Level have a menu bar with largely standardized menu options and corresponding pull-down menus.

- **Standard toolbar (only available in primary windows):**

The [Standard Toolbar \[Seite 307\]](#) contains pushbuttons for the basic R/3 functions such as OK, Save, navigation and help functions. The standard toolbar also includes the command field. The user can enter commands to the R/3 System in the command field.

- **Application toolbar (only available in primary windows):**

Underneath the standard toolbar, you see the Application Toolbar (Control Element). It contains application-specific functions which affect the entire screen.

- **Status bar (only available in primary windows):**

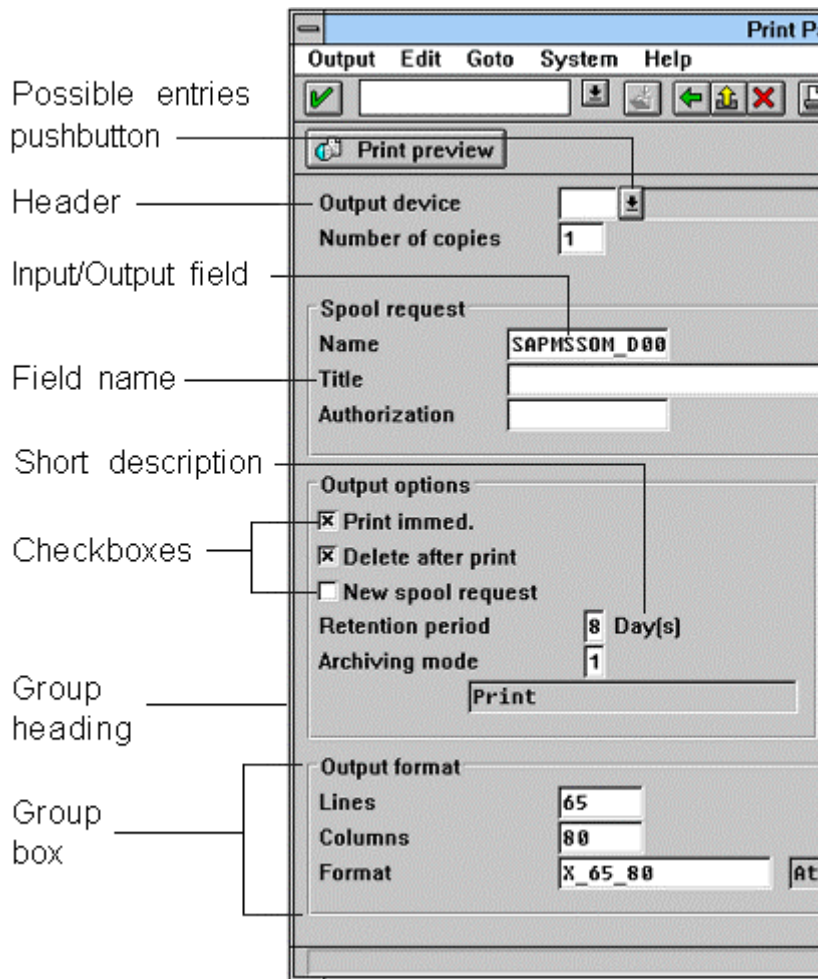
The [Status Bar \[Seite 305\]](#) is at the bottom of the window. The R/3 System displays messages in the status bar.

Work Area Design Elements

[Dialogfenster \[Seite 65\]](#)

[Drucktasten \[Seite 227\]](#)

The Window Work Area



Task data should be displayed clearly and arranged logically. The user should also be provided with appropriate control elements to edit data.

Work Area Control Elements

Data input and representation of information:

- **Data input/output:**

[Input/Output Fields \(Control Elements\) \[Seite 238\]](#) accommodate data and/or display data. [Field Names \(Control Element\) \[Seite 324\]](#) are placed in front of input/output fields.

Work Area Design Elements

Fields for which entry help is available can be followed by a [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#). A description (check text) can follow the I/O field and explain its contents.

- **Grouping information:**

Data in the work area of the window can be grouped together to form [Group Box \(Control Element\) \[Seite 280\]](#). Group boxes can be separated by blank lines.

- **Messages:**

Messages are displayed in a Dialog box or in the [Status Bar \[Seite 305\]](#).

- **Additional data:**

Additional data can be entered, edited or displayed in dialog boxes.

Editing data:

- **Actions:**

Pushbuttons With No Fixed Position (Control Element) in the Work area, but also in the standard and application toolbars trigger actions.

- **Selections:**

[Checkboxes \(Control Element\) \[Seite 176\]](#), [Radio Buttons \(Control Element\) \[Seite 193\]](#), [Single-selection and multiple-selection lists \[Seite 362\]](#) can be used to choose one or several options.

- **Navigation:**

[Scroll Bars \[Seite 202\]](#) and keys/buttons are available for navigation through tables, lists or large screens.

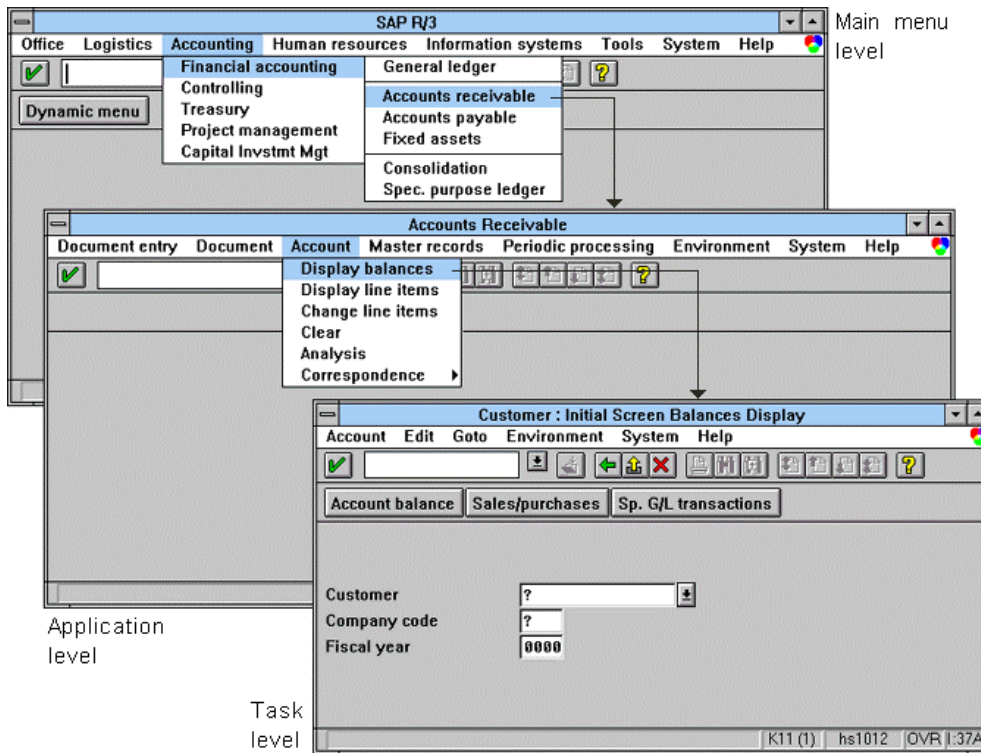
- **Table display:**

SAP tables ([Table Control \[Seite 340\]](#), step loops) and lists display data in tabular form and offer multiple processing options.

The Different Levels in the R/3 System

Levels in the SAP System

In the R/3 System, we distinguish between different levels and screens. This R/3 System structure should facilitate orientation and navigation in the system. The names assigned to the different levels and screens enable the user to understand the significance and contents of a window. This also supports developers when they classify screens of their own tasks, and it provides standard terminology for the documentation

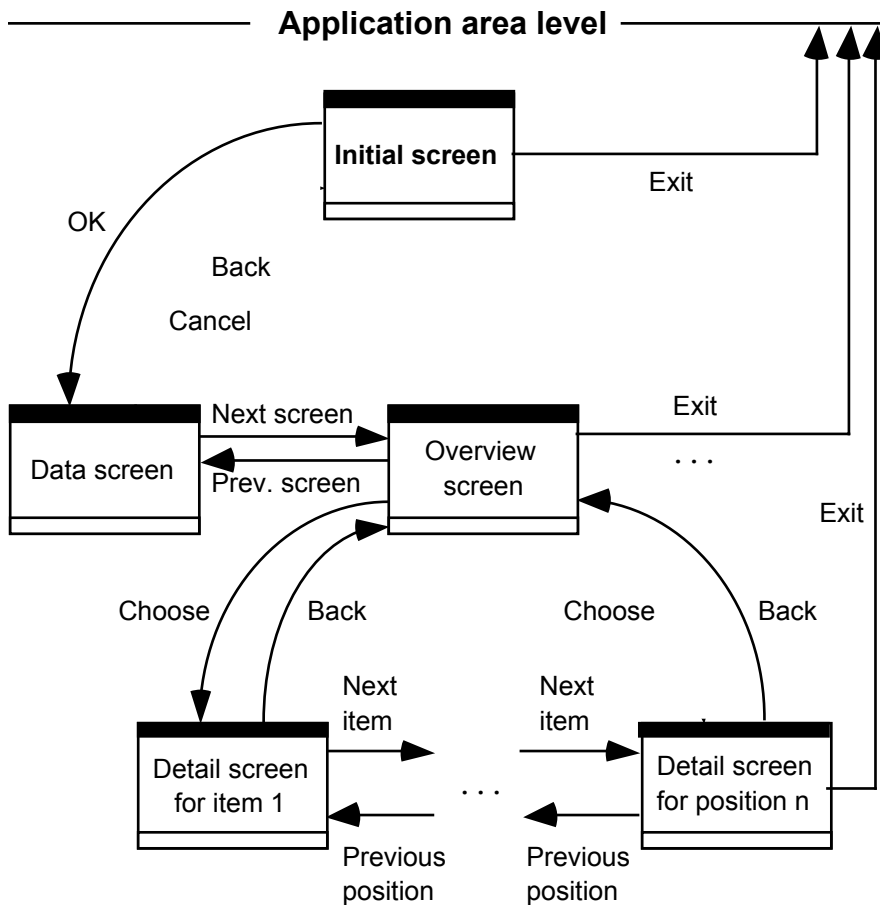


Navigation in the R/3 System

[Navigation Between Screens \[Seite 106\]](#)

means moving around in a software system, for example, from one task screen to the screen of another task. To perform any task in the R/3 System, the user has to choose an application from the Main Menu Level. This brings the user to the Application Level to choose an application task. At task level, the user navigates between object components, for example, between the header and data section of a document. An important feature are return navigation options: the system should allow the user to cancel or exit a task.

The Different Levels in the R/3 System



The user can combine functions and applications to create his or her own user menu. The user can specify tasks in the menu that he or she most often perform. The advantage of the user menu is that it lets the user directly access a particular task without navigating through the basic system hierarchy and a series of menus.

Developing Ergonomic Software with the R/3 System

[Object-Oriented Approach \[Seite 36\]](#)

[Identifying Basic Objects and Actions \[Seite 38\]](#)

[Outlining a Task: Hierarchy Structure and Work Processes \[Seite 39\]](#)

[Structure Conversion: Menu Structure \[Seite 40\]](#)

[Moving Around in the Structure: Navigation \[Seite 42\]](#)

[Template Design in the Screen Painter \[Seite 43\]](#)

[Details of the Interaction: Interaction Elements and Techniques \[Seite 44\]](#)

[At last: Let's talk about the Menu Painter! \[Seite 45\]](#)

[Important Additions to the Interaction: Help, Error and System Messages \[Seite 47\]](#)

[Summary \[Seite 48\]](#)

Object-Oriented Approach

Object-Oriented Approach

The user interface of the SAP R/3 System is based on the "object-oriented approach", also known as the object-action approach. The system presents a "world" of predefined things (objects) which the user can process in certain ways (actions). Ideally, this presentation uses the environment familiar to the user from previous work. The objects should not be abstract data structures but, for example, documents. Action names should not be technical but rather they should correspond to those of the real world environment (for example, *Post* instead of *Store*).

The object-oriented view of the real world has two advantages. On the one hand, it helps users find their way around when working with the computer. On the other hand, it also helps developers structure their tasks. This time, the task is not viewed from a programming perspective but rather from the perspective of the task to be performed.

The following sections explain how the object-oriented viewpoint can guide the software development process. The development of a task is split up into a sequence of steps in which specific attributes of the task and the corresponding user interface are determined. The focus is not on the programming problems to be resolved, but on how the task should be presented to the user. The following table illustrates the steps for the user interface. The individual steps are described in detail in the following sections.

Steps in the development of the user interface

No.	Procedure	Activities	Relevant Style Guide chapters
1	Identify basic objects	Determine objects and actions	Object-oriented user interface
2	Outline the task: Build the hierarchy structure and model the work processes	Determine task structure and processes	Object-oriented user interface
3	Convert the structure: Build the menu structure	Design menus (general and specific design)	Levels in the SAP System Menu chapter Overview of Standard Functions Overview of Menu Bars
4	Plan navigation structure	Determine pathways within tasks and in the system	Navigation functions - Overview Navigation between screens
5	Detail work: Design the templates in the Screen Painter	Design templates in the Screen Painter, design fields	Fields - Layout I/O fields, Field names
6	Details of the interaction: Interaction elements and techniques	Design interaction elements on templates in the Screen Painter	Control elements - Overview Sections on control elements Selection techniques Sections on tables

Object-Oriented Approach

7	Define menu structure in the Menu Painter	Define status in menu painter: menus, function keys and pushbuttons, fast access characters	Sections on menus Overview of Standard Functions Overview of Menu Bars
8	Important additions to the interaction	Determine help, messages, etc.	Dialog boxes General messages Message types and error groups Help, Possible entries pushbutton

Identifying Basic Objects and Actions

Identifying Basic Objects and Actions

The software developer's job is to model a business process in the software system. A number of methods are available supporting a systematic procedure. Using the object-oriented approach, the developer first determines the principle objects, object components and actions relevant to a task. This requires a good knowledge of the task's work processes and involvement of the user and specialists in the design process.



The developer is to create a mail system.

- **Objects:**

Folders: inbox, outbox, trash, bulletin board, private and shared folders, etc. mail documents, recipient lists, distribution lists etc.

- **Object components:**

Mail document: address section, text section, enclosures etc.

- **Actions:**

Create, send, change, resubmit, print, archive, delete a mail document

Relevant Section in the Style Guide

- [Object-Oriented User Interface \[Seite 129\]](#)

Outlining a Task: Hierarchy Structure and Work Processes

Once the objects and actions have been identified, they have to be structured in a meaningful way for the work process.

Work processes have to be analyzed. Often dependencies exist between objects or between actions on objects,. For example, some actions can only be executed after another action has been successfully completed. Some actions are only possible for certain objects or under specific circumstances. Very often procedures have to be restructured so that the computer can support them. This may result in work processes which have little in common with the original processes.

Once the work processes have been identified, they have to be subdivided into meaningful sequences of basic actions that can be carried out by a computer. This also requires a meaningful distribution of the work. The developer has to decide which actions should be carried out by the computer, which actions are to be performed by the user and which actions should be executed by the computer and the user together.

At this stage, the developer is actually structuring work processes and work places. Computer-based work places are different from the traditional work places. The new type of work can be demanding or ask too much of the user; it may also make not enough demands on the user. Users who used to work closely together with their colleagues may now be socially isolated because software programs allow them to perform all the work by themselves.

In this respect, developers have a very high responsibility as far as the user's work place is concerned!

Structure Conversion: Menu Structure

[Einstieg in eine Anwendung \[Seite 77\]](#)

[Menüs - allgemeine Gestaltung \[Seite 548\]](#)

Once the structure of a task and its processing have been specified, they have to be translated into a data processing solution. The R/3 System is menu-driven. The menu structure is therefore a very important and basic part of developing an R/3 task. It largely determines how easy it is for the user to choose and execute an action. It also has an impact on how easy or difficult it is for the user to switch from one action to another.

Levels in the SAP System and Navigation

The R/3 System is hierarchically structured. The user accesses the R/3 System at the main menu level. The user chooses an application from the first screen of the SAP System and then proceeds to the application level. For example, *Logistics* → *Materials Management*. The user then chooses an application task. The task could be, for example, to create a material master record. It is only at task level that the user enters and edits data, for example, materials.

The general system levels are predefined. It is the designer's job to provide the task structure which also has a hierarchical structure. The user needs to be able to navigate efficiently up and down the hierarchy and to use other cross-connections.

Application Menu Structure

The basic menu structure is largely standardized in the R/3 System. Thus, the developer should try to adapt his or her objects and actions to this structure. In doing so, the designer creates a "basic" menu structure of the task which is completed by specific functions. The homogeneity of the R/3 Systems depends upon the success of this process. Two things should be kept in mind here:

- Functions have to be assigned to particular menu bar options.
- When programming functions, you should use standard functions as far as possible because the user is familiar with these functions.

For more information on assigning functions to menus, please refer to Overview of Menu Bars. It contains the definitions of the menu options at task level. To successfully assign functions to the menu categories, you have to know the different menu definitions, for example, the different definitions for the *Goto*, *Extras*, and *Environment* menus.

When assigning functions to menus, you should check if existing standard functions are applicable. Some functions of a task are generally not task-specific and also occur in many other tasks, such as printing. Other functions, in turn, are task-specific. Please see Overview of Standard Functions for a list of the R/3 standard functions. Specific functions can often be performed by standard functions with only minor modification.

The result of this process is the complete task menu hierarchy. At this stage, you should already consider the task screen title bars. You can also determine which task processes are to be executed in primary windows and which are to be run in dialog boxes.

For more information on the menu structure of R/3 tasks, please refer to the following Style Guide sections:

Relevant Section in the Style Guide

In general

- [Levels in the SAP System \[Seite 72\]](#)
- Wildcard Characters

Menus

- General Design Guidelines
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- Menus: [<Application> \(Menu\) \[Seite 564\]](#), [Edit \(Menu\) \[Seite 565\]](#), [Help \(Menu\) \[Seite 570\]](#), [<Object> \(Menu\) \[Seite 574\]](#), [Goto \(Menu\) \[Seite 580\]](#), [Environment \(Menu\) \[Seite 583\]](#), [Extras \(Menu\) \[Seite 585\]](#)
- [Overview of Menu Bars \[Seite 679\]](#)

Functions

- [Functions - Overview \[Seite 451\]](#)
- [Overview of Standard Functions \[Seite 646\]](#)

Moving Around in the Structure: Navigation

Moving Around in the Structure: Navigation

The menu structure is only a collection of possible activities. Once the hierarchy of a task has been designed, the designer has to consider how the user should navigate through this structure and where particular work processes should take place in the structure. In a building, for example, certain activities take place in certain rooms. On the computer, all the action takes place on screen templates. The designer does not have to come up with details of the screen design at this point yet.

For the moment, the designer should determine which actions are to be executed on which screens and in which order. At this stage, you can already determine which task processes are to be executed in primary windows and which are to be run in dialog boxes. Once the tasks for the individual screens have been identified, screens should be named according to title bar design guidelines (at this point, still on the paper manuscript).

The next step is to determine the hierarchy and/or the screen sequence. Many task processes require more than one screen in order to process the relevant data. Other tasks consist of several subtasks between which the user has to switch back and fourth. The order of the screens can be either made mandatory or left to the discretion of the user. A tip for this planning stage: write the title of each screen on a piece of paper and rearrange the screens as often as required to come up with a logical order of the screens.

The following questions may arise during this process:

- Which pathways does the user have to take to accomplish the task?
- How is the user to proceed when a task terminates abnormally?
- Which return options does the user have?
- Where does the user arrive in the system when he or she exits certain functions?

All possible routes and corresponding navigation functions can then be identified.

For more information on navigation issues, please refer to the following sections:

Relevant Section in the Style Guide

- [Functions - Overview \[Seite 451\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Navigation Between Screens \[Seite 106\]](#)
- [Title Bar \[Seite 152\]](#)

Template Design in the Screen Painter

After the structure of the task has been determined, you can use the Development Workbench tools to design the individual screens and templates, or to be more precise, to design the work area. You define the screen layout in the Screen Painter, the menu structure and the function key setting etc. in the Menu Painter. It is up to the designer's personal preferences which job he or she does first. Usually the work in the Screen Painter and Menu Painter goes hand in hand.

In the Screen Painter you have to determine which information belongs on a screen, how the corresponding fields and field names are distributed, and how they can be meaningfully combined to form groups with related contents.

The basic design guidelines are:

- simplicity
- clarity
- transparency

Do not overload the work area with information but use a clear structure to ensure that information can be retrieved quickly and processed efficiently. If in doubt, use the following principles:

- One screen (template) should be assigned to one task (less is more).
- Efficient task handling has priority over an aesthetically pleasing arrangement.

You structure the templates for your task in the Screen Painter. It might be useful though, to sketch your templates on paper first.

For more detailed information, please refer to the following Style Guide sections:

Relevant Section in the Style Guide

- [Abbreviations \[Seite 593\]](#)
- [Control Elements - Overview \[Seite 168\]](#)
- [Effects on Perception \[Seite 89\]](#)
- [Field Names \(Control Element\) \[Seite 324\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Formulations \[Seite 601\]](#)
- [Group Box \(Control Element\) \[Seite 280\]](#)
- [Highlighting \[Seite 603\]](#)
- [Hyphenation \[Seite 620\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Uppercase/Lowercase \[Seite 619\]](#)

Details of the Interaction: Interaction Elements and Techniques

Once the fields have been positioned, that is, the screen layout has been more or less completed, the object-ecion approach requires designers to implement further control elements in the work area of the template. Designing the exact interaction processes implies taking into account which interaction elements and techniques are to be used for which tasks.

Control elements such as check boxes and radio buttons serve a particular purpose, that is, choosing one or more options from several possibilities. Control elements follow a certain behaviour and also require a particular user behaviour. Designers have to observe important Style Guide conventions for certain control elements such as checkboxes, radio buttons, scroll bars, menus, or tables and lists.

The Style Guide also provides information on the following topics:

Relevant Section in the Style Guide

Interaction elements

- [Control Elements - Overview \[Seite 168\]](#)
- [Checkboxes \(Control Element\) \[Seite 176\]](#)
- [Radio Buttons \(Control Element\) \[Seite 193\]](#)
- [Tables and Lists \[Seite 339\]](#)

Important methods

- [Choose \(Function\) \[Seite 487\]](#)
- [Selection Methods - Overview \[Seite 95\]](#)
- Tables: [Tables - Scrolling \[Seite 376\]](#), [Tables - Maintenance \[Seite 383\]](#)

At last: Let's talk about the Menu Painter!

[Funktionen - Übersicht \[Seite 451\]](#)

[Funktionen - Übersicht \[Seite 451\]](#)

[Menüs - allgemeine Gestaltung \[Seite 548\]](#)

By now, the design process has progressed to the point where you can proceed to define statuses in the Menu Painter. A status comprises functions in menus, print and functions keys, access characters and title texts which are assigned to a task in a particular situation. If a task requires different settings, they have to be defined individually. The task must switch between statuses, if required.

At this stage, the designer determines the menu structure on the computer, assigns function keys, completes the application toolbar, and assigns access characters.

The Menu Painter can be particularly useful because it makes suggestions and allows you to check if your final result conforms to the R/3 interface standards. For a summary of all elements, please refer to the sections [Overview of Standard Functions \[Seite 646\]](#), [Overview of Menu Bars \[Seite 679\]](#) and [Functions - Alphabetical List](#).

Check reports can also be quite useful, although they are not included in the standard menus. You can call them with transaction code ZCUA. To check the various aspects of the user interface, you can use the menu option *User interface* which offers approx. ten different check reports.

Relevant Section in the Style Guide

Functions, Function Keys, Pushbuttons

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Defining Functions \[Seite 456\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Functions - Overview \[Seite 451\]](#)
- Pushbuttons With No Fixed Position (Control Element)
- [Standard Toolbar \[Seite 307\]](#)

Menus

- [Access Characters \[Seite 184\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- General Design Guidelines
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- Menus: [<Application> \(Menu\) \[Seite 564\]](#), [Edit \(Menu\) \[Seite 565\]](#), [Help \(Menu\) \[Seite 570\]](#), [<Object> \(Menu\) \[Seite 574\]](#), [Goto \(Menu\) \[Seite 580\]](#), [Environment \(Menu\) \[Seite 583\]](#), [Extras \(Menu\) \[Seite 585\]](#)

At last: Let's talk about the Menu Painter!

Overview

- Functions - Alphabetical List
- [Overview of Standard Functions \[Seite 646\]](#)
- [Overview of Menu Bars \[Seite 679\]](#)

Important Additions to the Interaction: Help, Error and System Messages

The task design is still not complete as far as the user interface is concerned. The users require additional support for their work, for example, information on

- the progress of the processing
- error messages
- remedies for error situations.

The R/3 System offers various forms of user support:

- the F1 help explains the meaning of fields
- the F4 help provides the possible input values for a field
- Set/Get parameters and other features remember user input or set default values.

The developer does not always need to enter these or other information himself.

Error messages are an especially critical factor. They should contain information on the cause of the error and possible correction action, if required. The formulation of the message is therefore very important. Avoid technical descriptions or explanations, and use straightforward, easily understood terminology. Also avoid phrasing that blames the user or implies user error. Otherwise, the message might have a negative effect on the user.

For more information on user support, please refer to the following sections:

Relevant Section in the Style Guide

- [Modal Dialog Box \[Seite 65\]](#)
- [General Messages \[Seite 604\]](#)
- [Message Types and Error Groups \[Seite 611\]](#)
- Help in particular: [Help \(Function\) \[Seite 509\]](#), [Possible Entries \(Function\) \[Seite 499\]](#), [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#)

Summary

Summary

Because of the complex nature of R/3 user interface design, this guide cannot provide information on every possible interface issue. Nevertheless, the authors hope that it supports developers and designers in planning and structuring R/3 tasks and that the Cross-References to specific chapters and sections in the Style Guide help to use it more efficiently.

Some of our long-term goals formulated in the previous version of the Style Guide have now been achieved:

- to improve the availability of the Style Guide, it is available in the online system which provides quick and effective access
- ergonomic rules contained in the Style Guide have been incorporated as checks in the development environment: the Menu Painter offers extensive support to the developer for the definition of menus and function keys. In the Screen Painter, a number of consistency checks are carried out to support the developer.

Since the SAP GUI is constantly developed further, new elements have to be added with every new release of the system. This also means that new guidelines have to be added to the Style Guide. It cannot be our aim to regulate the development of user-friendly applications by constantly creating new rules and regulations. Instead, we have to find ways and means to reduce the time needed to develop the user interface. In this respect, the style guide is not intended to hinder, but rather to facilitate the designer's work.

One step in this direction are the new example transactions. They illustrate the most important screen and list design guidelines. Developers can access these transactions and learn about the guidelines from the examples. The Style Guide could then be the final authority with regard to ergonomic issues and become a systematic reference book.

General Design Guidelines

This chapter explains the basic guidelines to be observed when designing the SAP GUI. These guidelines are arranged in alphabetical order by keyword. The topics cover the menu structure of the R/3 System, navigation within the system, important interaction techniques, and so on.

[Accessing Tasks \[Seite 77\]](#)

[Effects on Perception \[Seite 89\]](#)

[Exclusive OR Selections on Initial and Selection Screens \[Seite 84\]](#)

[Levels in the SAP System \[Seite 72\]](#)

[Modal Dialog Box \[Seite 65\]](#)

[Navigation Between Screens \[Seite 106\]](#)

[Object-Oriented User Interface \[Seite 129\]](#)

[Prompts in Dialog Boxes \[Seite 50\]](#)

[Referencing Objects \[Seite 134\]](#)

[Searching in Structured Data \[Seite 142\]](#)

[Selection Methods - Overview \[Seite 95\]](#)

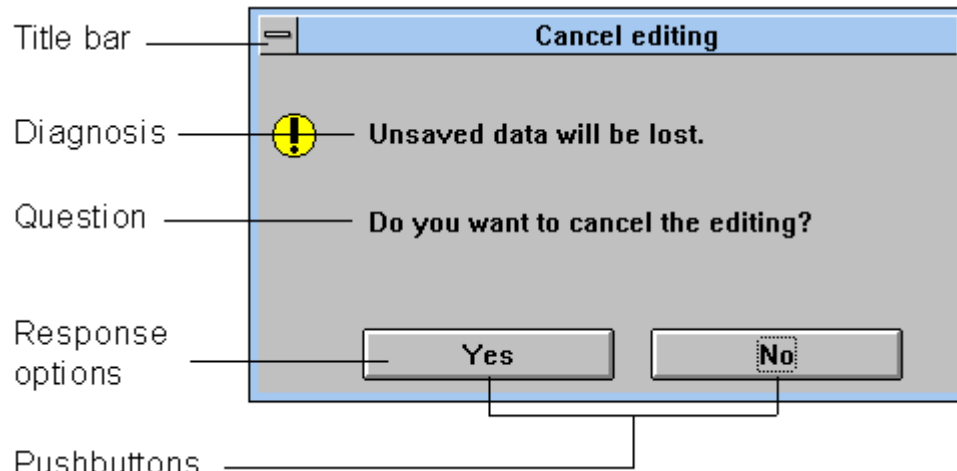
[Title Bar \[Seite 152\]](#)

Prompts in Dialog Boxes

Prompts in Dialog Boxes

If the user performs an action with potentially destructive or irreversible consequences, or it is likely that the action was performed by mistake, the system should display a dialog box explaining the implications and offering alternative courses of action, including cancelling the action.

Illustration



The elements of a dialog box

[Categories \[Seite 51\]](#)

[General Layout Guidelines \[Seite 52\]](#)

[Confirmation Prompts After Returns \[Seite 53\]](#)

[Confirming Actions \[Seite 56\]](#)

[Confirming Actions After Previous Message \[Seite 59\]](#)

[Choosing Between Two Related Alternatives With or Without Previous Message \[Seite 60\]](#)

[Technical Notes \[Seite 61\]](#)

[Cross-References \[Seite 64\]](#)

Categories

Three categories of prompts are distinguished:

- Prompts which the system displays every time a user action may cause a loss of data - so-called **confirmation prompts**
- Prompts which require the user to **confirm** an action
- Prompts which require the user to **choose** between two alternatives.

Within these categories, designers can use various forms of display and modes of operation, all of which are implemented in **function modules**. These function modules are always specified in uppercase letters to the left of the allocated text. The "Technical Notes" describe the use of each function module listed.

In the implementation, a distinction is made at certain points between **destructive** and **non-destructive** actions or response options. "Destructive actions or response options" result in a loss of data. If no data is lost, they are referred to as "non-destructive actions or response options".



The messages mentioned above should not be confused with E-, W-, I-, S- and A-type messages which are maintained in table 100 (see [Message Types and Error Groups \[Seite 611\]](#)).

Confirmation Prompts (with Returns/Exits/Navigation)

The system displays **confirmation prompts** when the user executes one of the following functions by activating function keys, pushbuttons or menu options:

- F15=EXIT or the *Exit* pushbutton in the standard toolbar or the corresponding menu option
- F3=BACK or the *Back* pushbutton in the standard toolbar or the corresponding menu option
- F12=CANCEL or the *Cancel* pushbutton or the corresponding menu option
- F17=OTHER <OBJECT> or the corresponding menu option
- OK-CODE=N
- another object class from the application area menu
- another action from the <object> menu

Other System Prompts

The system requests the user, with or without messages, to confirm an action, for example:

- if entities - for example files, folders, letters, tables, and so on - are to be deleted,
- if entities are to be copied,
- if a decision has to be made about how to continue processing.

General Layout Guidelines

General Layout Guidelines

Dialog boxes contain the following elements:

- A **title bar** must be defined in dialog boxes according to the title bar design guidelines (see [Title Bar \[Seite 152\]](#)).
- A diagnosis text might be displayed in the dialog box, explaining why the following question is asked.
- A **question** refers to the action previously performed, the possible loss of data, or the diagnosis text.
- The "Yes" and "No" **responses** are displayed as two **pushbuttons** (exception: see Decision Between Two Related Alternatives in this keyword section). The "Yes" response should be presented first.

The cursor is generally placed on the non-destructive option.



The user responds to the question posed in the dialog box by moving the cursor with the **TAB** key to the desired option and then pressing **ENTER** (or by double-clicking on the desired option). It can also be chosen by means of a single mouse click.

- The **Cancel pushbutton** is also displayed. It is positioned to the right of the response buttons.
- **Blank lines** between the individual elements of the dialog boxes are set automatically by the system prompt function modules.

Using Icons in Dialog Boxes with Confirmation Prompts

The standard function modules relating to confirmation prompts are automatically assigned to the corresponding icon:

- **POPUP_TO_CONFIRM_LOSS_OF_DATA**
This confirmation prompt is displayed when the user executes the *Cancel* function. The prompt is assigned an exclamation mark icon to indicate that data may be lost.
- **All other prompts**
All other prompts are assigned an question mark icon to indicate that the user has to make a decision.

	WARNING	Loss of data		QUESTION	All others
---	---------	--------------	---	----------	------------

Icons for safety or confirmation prompts

If you design your own prompts, place the corresponding icon to the left of the prompt text (centered vertically).

Confirmation Prompts After Returns

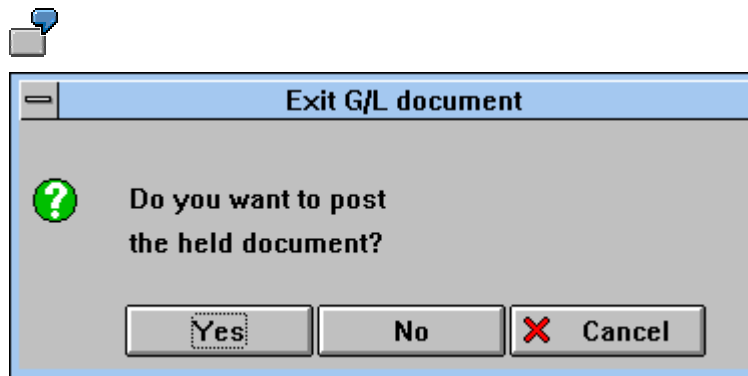
Confirmation prompts are displayed when leaving the initiated action may cause a loss of data. There are two possibilities:

- Return with Post/Save: The user is able to save the entered data and then returns to the previous stage in the function.
- Return without Post/Save: The user is not able to save the entered data, only return to the previous stage in the function.

With Post/Save

("POPUP_TO_CONFIRM_STEP" or "POPUP_TO_CONFIRM_WITH_VALUE")

The first possibility should be the **standard return option**, if possible. The user leaves the task function, and the previously entered data is either posted or saved, otherwise unsaved data will be lost. In addition, the user can use the "[Cancel \[Seite 470\]](#)" option to close the dialog box again and cancel the initiated action.



Dialog box with confirmation prompt when *Post/Save* is possible (*Exit*)

Procedure

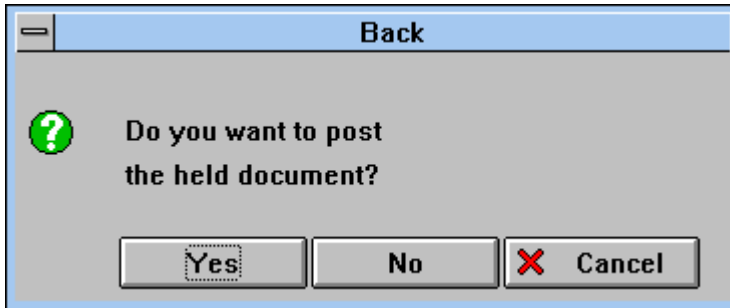
- The user has activated the [Exit or End <application> \(Function\) \[Seite 492\]](#) function.
- A dialog box is displayed because data has already been entered or changed.

Layout

- The question refers to the possible loss of data.
- The non-destructive option "Yes" is selected - that is, the cursor is positioned on it. To post data, the user chooses "Yes"; to leave the task without posting, the user chooses "No".

Confirmation Prompts After Returns

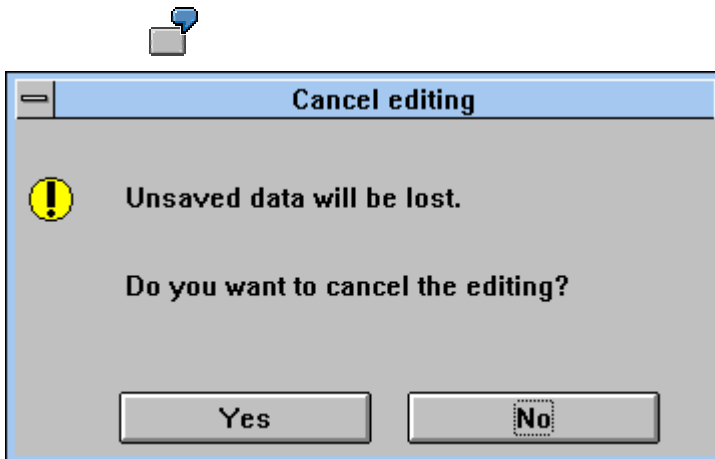
Further Example



Dialog box with confirmation prompt when *Post/Save* is possible (*Back*)

Post/Save is Not Possible ("POPUP_TO_CONFIRM_LOSS_OF_DATA")

The second return option is used if data is inconsistent or if too extensive costly checks would need to be performed. The user can respond to the confirmation prompt in one of two ways: Either leave the task function and lose the previously entered data, or return to the task to proceed.



Dialog box for inconsistent data which cannot be saved

Procedure

- The user has activated the [Cancel \[Seite 470\]](#) function.
- A dialog box is displayed because data has already been entered or changed.

Layout

- A diagnosis text is displayed. In this case, it always reads:
Unsaved data will be lost.
- The question refers to the initiating action.
- The "No" response option is selected.

Confirming Actions

Confirming Actions

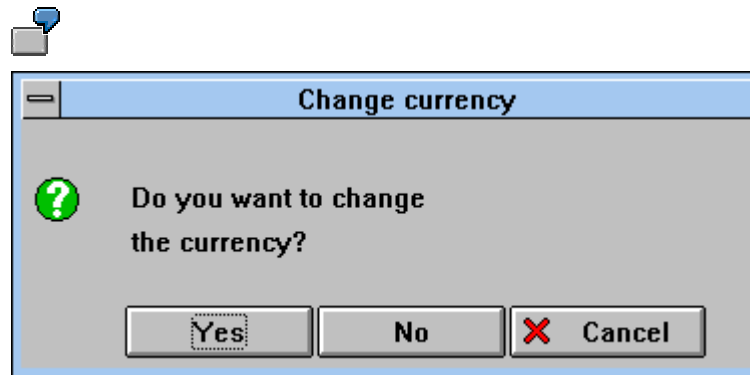
There are two kinds of dialog box which require the user to confirm an action:

- The user is to confirm a non-destructive action.
- The user is to confirm a destructive action.

Non-Destructive Action

(**POPUP_TO_CONFIRM_STEP** or **POPUP_TO_CONFIRM_WITH_VALUE**, Pre-Selected Response: "Yes")

In this case, you should consider whether the dialog box is really necessary, as it will slow down processing.



Dialog box for confirming a non-destructive action

Procedure

- The user has activated the *Change Currency* function.
- A dialog box is displayed because the function has to be confirmed here.

Layout

- The question refers to the initiating function.
- The "Yes" response option is pre-selected.

Further Examples of Questions

- Do you want to copy the document?
- *Do you want to change the rules for determining the account?*

Destructive Action

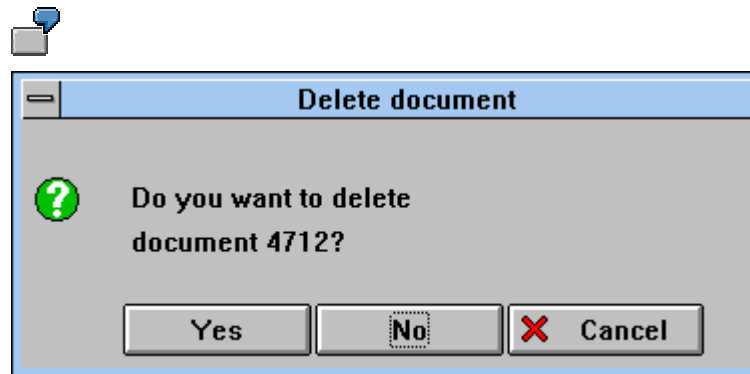
(POPUP_TO_CONFIRM_STEP or POPUP_TO_CONFIRM_WITH_VALUE Pre-Selected Response: "Yes" or "No")

If the destructive operation may have irreversible consequences, a dialog box should always be displayed. For example, the user has selected 20 lines in a list and then chooses [Delete \[Seite 515\]](#).

Should Response Option "Yes" or "No" be Selected?

If a transaction offers several operations of equal priority which require the user to confirm the action, and some of these operations are of a destructive and others of a non-destructive nature, the "Yes" option can be the pre-selected response for all operations. In these unusual circumstances the "Yes" option is pre-selected for a destructive action.

If, however, the destructive operation is the less frequently used function, then the "No" response should be pre-selected to provide for maximum protection against a loss of data.



Dialog box for the confirmation of a destructive action

Procedure

- The user has activated the *Delete Document* function.
- A dialog box is displayed since the function has to be confirmed.

Layout

- The question refers to the initiating function.
- In the example, the "No" response option is pre-selected.

Further Examples of Questions

- Do you want to delete the marked document 4711?
- *Do you want to delete the existing components first?*
- *Do you want to delete the selected items?*
- *Do you want to delete the documentation?*

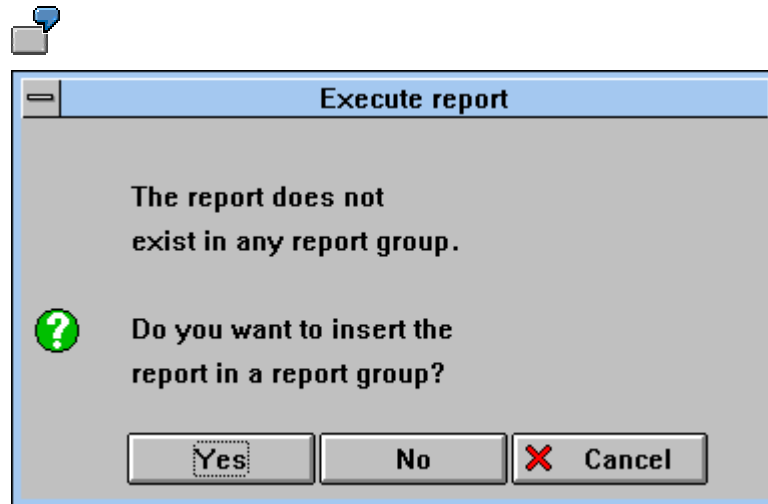
Confirming Actions

Confirming Actions After Previous Message

The options described in the previous section can be additionally supplemented with a **warning**, **information** or **diagnostic message** which precedes the actual question.

Message + Action (POPUP_TO_CONFIRM_WITH_MESSAGE)

First, the system displays a dialog box with a message specifying the critical processing situation. Then the user is asked how he or she wants to proceed.



Dialog box for confirming an action after previous message

Layout

The layout is the same as that described in the previous section on dialog boxes with a message text (concluded with a full stop) placed in front of the question.

Further Examples of Diagnosis + Question

Diagnosis: *You have changed the exclusion indicator.*

Question: *Do you want to copy the new indicator into the current records?*

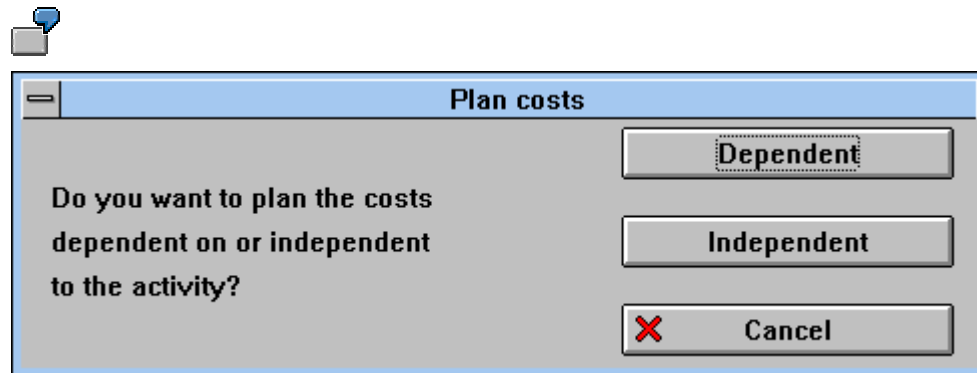
Choosing Between Two Related Alternatives With or Without Previous Message

Choosing Between Two Related Alternatives With or Without Previous Message

The most general form of a dialog box containing a prompt is the dialog box which requires the user to choose between **related alternatives**. It can be preceded by a message or warning or diagnostic information.

(Message) + Decision (POPUP_TO_DECIDE and POPUP_TO_DECIDE_WITH_MESSAGE)

Sometimes the system asks the user an either/or question which does not allow yes/no answers. In this case, the alternatives available to the user must be clear. The system can display a message first.



Dialog box with two related alternatives

Layout

- In the question, the two options are linked with "or".
- Next to the question, separated by a blank line, two alternative responses and a cancelling option are presented in a horizontal button arrangement.

Further Examples of Diagnosis + Question + Response Alternatives

Diagnosis: *The selected chart of accounts is already active.*

Question: *Do you want to exit the activation procedure or deactivate the chart of accounts?*

Response options: Exit/Deactivate

Technical Notes

General Information

Dialog boxes with a confirmation prompt should only be displayed if data entered may be lost. You can use the system flag SY-DATAR to check whether this is the case. When the screen is loaded, the system resets the flag to "SPACE" (not in the case of dialog boxes displaying messages). If the user enters or changes data in a field without previously pressing ENTER, it is set to "X". It can be tested at PAI. Depending on this flag, it is possible, for example, to decide within a module "x" defined with MODULE x AT EXIT COMMAND whether a confirmation prompt is to be displayed or not. If ENTER was pressed at some point, the task has to record whether data has already been changed in a local flag.

Function Modules

The developer can use **function modules** for the various prompt categories (see documentation under transaction "UNIT", function group "SPO1"). The following sections explain the parameters, the return values, and the functionality of the various function modules.

All function modules position the corresponding texts and pushbuttons at the locations provided in the dialog box, pre-select certain response options, guarantee the selection logic and center the dialog box in a predefined size within the task window, unless the developer specifies otherwise.

Parameters

Prompt

The prompt is transferred to the function module in two separate lines! This may cause problems during translation if the text is stored under two different numbers in the text pool. For this reason, the number for one test should at least be consecutive.

Title Bar

A **title bar** must be transferred to all function modules listed below. The title bar should contain the affected <object> and the chosen <action>, if possible (see [Title Bar \[Seite 152\]](#)).

POPUP_TO_CONFIRM_STEP

Transfer Values

- A **question** of no more than two lines of <= 35 characters is to be formulated as follows: "Do you want to <action> <object> first?". For example: "Do you want to post the entered document first?".
- A **parameter** "Y" or "N" specifying whether "Yes" or "No" is the pre-selected option (default is "Y")

Functionality

The module positions the variable question and the response alternatives in the dialog box. The option transferred as a parameter is pre-selected.

Technical Notes

Return Value

"Y" for "Yes", "N" for "No", and "C" for "Cancelled"

POPUP_TO_CONFIRM_WITH_VALUE

This function module is identical to the function module POPUP_TO_CONFIRM_STEP except for the following features:

The explicitly named object can also be transferred as a parameter. Instead of "Do you want to post the entered document first?", the question is then: "Do you want to post document 4711 first?".

POPUP_TO_CONFIRM_LOSS_OF_DATA

Transfer Values

The **question** has the syntax: "Do you want to <previously selected action> <object>?". For example: "Do you want to cancel the transfer of primary costs?" or "Do you want to return from the list screen?"

The question is transferred in no more than two lines of <= 35 characters, starting with "Do you want to" and ending with a"?".

Functionality

Besides the variable text, the module displays the predefined diagnosis text and the response options ("No" is pre-selected) in the dialog box.

Return Value

"Y" for "Yes", "N" for "No", and "C" for "Cancelled"

POPUP_TO_CONFIRM_WITH_MESSAGE

The following values are transferred

- **Message / diagnosis / information**, terminated by a period, in three lines of <= 35 characters
- the **question** in two separate lines of <= 35 characters, starting with "Do you want" or "Should <object> <action> ? "
- a **parameter** specifying that "No" is the pre-selected response option.

Functionality

The module displays the variable message, the variable question and the response options ("No" can be pre-selected) in the dialog box.

Return Value

"Y" for "Yes", "N" for "No", and "C" for "Cancelled"

POPUP_TO_DECIDE and POPUP_TO_DECIDE_WITH_MESSAGE

The following values are transferred

- The **message / diagnosis / information** is transferred with POPUP_TO_DECIDE_WITH_MESSAGE, terminated by a period in three lines of <= 35 characters.
- The **question** is transferred to both function modules in two separate lines of <= 35 characters, starting with "Do you want" or "Should <object> <action> or <object> <action> ?" (the constituent parts of the question are linked with "or". Both alternative responses are specified).
- Two explicit **alternative responses** and information about which of them is pre-selected (possibly a number) are transferred to both modules. The alternative responses should consist of **one** word, if possible. The designer has to decide which alternative is pre-selected (generally the more frequent or less critical option).

Functionality

The modules display the variable message, the variable question and the variable alternative responses in the dialog box and pre-select the specified alternative. The location of the dialog box within the window is predefined.

Return Value

"1" for the first alternative, "2" for the second, and "C" for "Cancel".

Icons

Message and confirmation prompts are defined as one line, however, they occupy three lines and are four characters wide.

Cross-References

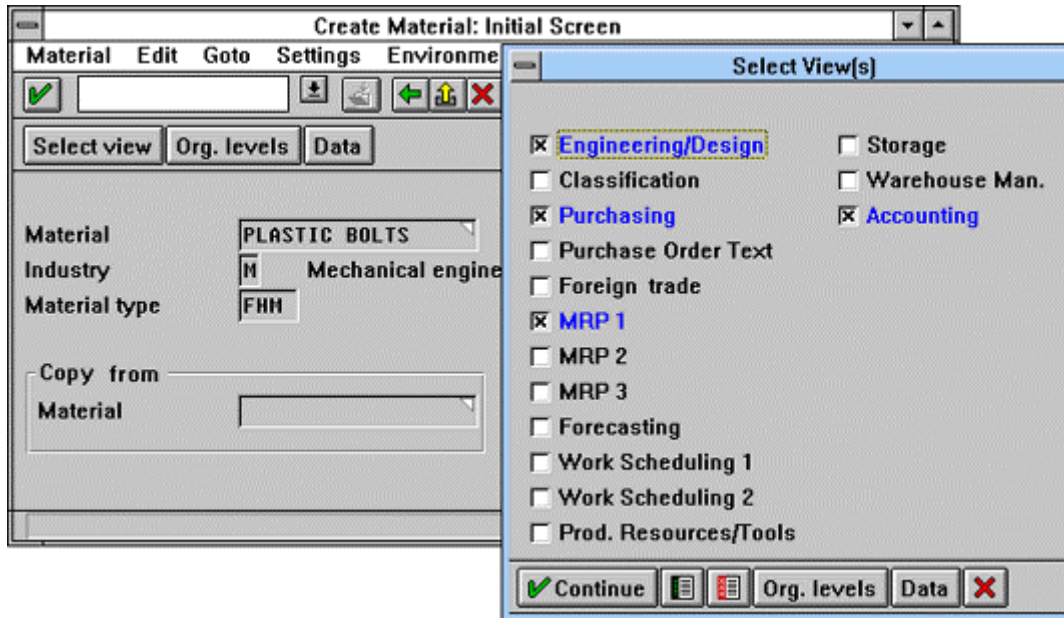
Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Formulations \[Seite 601\]](#)
- [General Messages \[Seite 604\]](#)
- [Message Types and Error Groups \[Seite 611\]](#)
- [Title Bar \[Seite 152\]](#)
- [Uppercase/Lowercase \[Seite 619\]](#)

Modal Dialog Box

A dialog box is classified "modal" if the user must respond to the dialog before continuing work in another window (generally the primary window).

Illustration



Primary window with a modal dialog box

[Guidelines \[Seite 66\]](#)

[Pushbuttons or Function Keys in Dialog Boxes \[Seite 68\]](#)

[Technical Notes \[Seite 70\]](#)

[Cross-References \[Seite 71\]](#)

Guidelines

Guidelines

Uses

- The current main action within a transaction runs in primary windows. For every main action, at least one primary window has to be defined to carry out the *Save* or *Post* action.
- Dialog boxes supplement the main action by displaying additional information/data and/or allowing further entries/functional specifications/selections including in required-entry fields.
- There must be no **sequence** of more than three dialog boxes on the screen. A dialog box initiated with `F4=Possible entries` may be displayed additionally.

Menu Bar in Primary Window and Dialog Box

The menu bar in the primary window is locked if a modal dialog box is active on the screen. The dialog box has no menu bar of its own.

Error Messages in the Dialog Box

If the entries of a dialog box are processed (for example, after pressing `ENTER`), the system can display an error message on recognition of an error. The message is displayed in another dialog box unless it is an S message (see [Message Types and Error Groups \[Seite 611\]](#)). The user then has the option of correcting the input value in the dialog box - that is, the dialog box remains where it is and the user receives the request to correct the error (or he cancels with `F12`).

Navigation in Dialog Boxes

- line by line or character by character with the arrow keys
- with the `TAB` key from one input field to the next
- with the mouse:
 - Single click = Cursor positioning
 - Double click = Cursor positioning + Select + `ENTER`
(currently corresponds to `F2=CHOOSE`)
- When the dialog box is displayed, the cursor should be placed on the first field to be selected or in the first entry field. If, as for example in the display-only mode, no entry fields exist, the cursor should be placed on the first displayed field which can contain variables. The cursor is to be set correspondingly.

Positioning Dialog Boxes

Adhere to the following guidelines:

- The first dialog box to be displayed should, if possible, be positioned so that the relevant information of the primary window is not covered up.
- If this is not possible, the dialog box should be centered on the screen if no other dialog boxes can be displayed.
- If possible, the dialog box should not be displayed at the bottom right or left of the screen (squeezed up against the edge).

Guidelines

- Further dialog boxes to be displayed are positioned so that the title bar of the previous dialog box is not covered up - that is, moved down at least two lines. In addition, they are moved a few characters to the left or to the right.

Pushbuttons or Function Keys in Dialog Boxes

Pushbuttons or Function Keys in Dialog Boxes

Pushbuttons should be lined up across the bottom of the dialog box:

•	ENTER=<ACTION>	(always the first button)
	FX=SPECIAL FUNCTIONS	(ascending order)
	F12=CANCEL	(the last developer-definable key)
	(In this order)	

- The buttons
 - **ENTER=<ACTION>**
 - **F12=CANCEL**
 must always be displayed. Abbreviations are allowed (see [Abbreviations \[Seite 593\]](#)).
- The following buttons should not be displayed:
 - scrolling possibilities:
 - F22=PREVIOUS PAGE (see [Previous Page, Next Page \(Functions\) \[Seite 535\]](#))
 - F23=NEXT PAGE (see [Previous Page, Next Page \(Functions\) \[Seite 535\]](#))
 - F4=POSSIBLE ENTRIES (see [Possible Entries \(Function\) \[Seite 499\]](#))
 - F2=CHOOSE (see [Choose \(Function\) \[Seite 487\]](#))
 - F9=SELECT (see [Select/Deselect \(Function\) \[Seite 516\]](#))

Executing Dialog Boxes

Executing a dialog box (that is, activating the **ENTER** key) is possible via the first pushbutton **ENTER=<ACTION>**: normally **<ENTER=EXECUTE>** (see [Execute \(Function\) \[Seite 485\]](#)), **ENTER=CONTINUE** (see [Continue \(Function\) \[Seite 540\]](#)) or after pressing **F4** from the primary window or for dialog boxes which contain a selection list, also **<ENTER=CHOOSE>**.



In this case, **F2** additionally has to be reserved for [Choose \(Function\) \[Seite 487\]](#) so that double-clicking with the mouse is also supported; **F2**, however, is not displayed.) Otherwise, the function to be executed can also be referred to directly ([Copy \(Function\) \[Seite 513\]](#)), for example).

Exception:

Dialog boxes which contain a question are instead provided with their own pushbuttons for the positive and the negative response. The **ENTER** key is then not displayed. Messages from the system which have to be confirmed receive the name **ENTER=CONFIRM**, in the case of read-only dialog boxes, **ENTER=CONTINUE** (see [Prompts in Dialog Boxes \[Seite 50\]](#)).

Pushbuttons or Function Keys in Dialog Boxes**ENTER Key**

The dialog box and the entries made are processed if the user presses the `ENTER` key. Then, the system either returns to the previous primary window or navigates to a new one or another dialog box is displayed (in this case the previous dialog box remains on the screen).

F12=Cancel

If `F12=CANCEL` is pressed, the dialog box disappears and the program either returns to the primary window or to the last dialog box (even if the dialog box contains required entry fields).

If possible, the previous status is fully restored - that is, the entries are "forgotten" and any processed entries are cancelled.

In extraordinary circumstances, if the entries can no longer be cancelled, either a dialog box is displayed with a warning that certain processing steps have already been carried out, or an error message is displayed in a dialog box informing that the "Processing step cannot be cancelled completely". To leave the processing, the user has to press `ENTER`. `F12` returns to the last processing step.

Read-only dialog boxes or dialog boxes displaying information are an exception. `F12=CANCEL` is actually reserved here, but it is not displayed.

F1=Help

`F1` is always active and is associated with individual fields, as in the primary window. Whether `F1` is displayed in the application toolbar depends on the global configuration and cannot be influenced by the developer; `F1` is currently not displayed.

F4=Possible Entries

`F4` is **not** displayed in the application toolbar.

After pressing `F4`, another dialog box is displayed (the first remains where it is) with the respective possible entries.

F3=Back, F15=Exit

In dialog boxes, `F3` and `F15` **are not allowed**. Dialog boxes are side paths which can only be cancelled with `F12`.

Save

Saving via `F11` is **not allowed in modal dialog boxes**. The *Save or Post* action always has to be carried out in a primary window. If a dialog box with a confirmation prompt (for example, after pressing `F15=EXIT`) of the lines *Should data be saved?* is displayed, the user can save by answering in the affirmative. However, data cannot be directly saved with `F11=SAVE`.

Technical Notes

- Modal dialog boxes are initiated with the ABAP/4 keyword CALL SCREEN STARTING AT X ENDING AT Y. To do this, the flag for the modal dialog box has to be set in the screen attributes in the Screen Painter.
- Titles and function keys are defined in the Menu Painter. In the program, the title must be set with SET TITLEBAR and the status with SET PF-STATUS. The title is then automatically centered in the title bar of the window. The title is not underlined. In the Screen Painter, the title bar is not to be taken into account during the definition of the window. On output, the dialog box is automatically extended by one line.
- The F12 functionality (*Cancel*) has to be programmed by the developer himself.
- The pushbuttons are defined as function keys in the Menu Painter. The ENTER key can and must be reserved (function key <00>, any function code). The function key F1 must not be reserved. The text in pushbuttons is followed by an ellipsis (...) if it initiates another dialog box. Programmers have to set the ellipsis themselves.

Cross-References

- [Control Elements - Overview \[Seite 168\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Message Types and Error Groups \[Seite 611\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Title Bar \[Seite 152\]](#)

Levels in the SAP System

Levels in the SAP System

A program system should have a structure that provides the user with an appropriate area in which to accomplish a given task. On the one hand, this area should be as small as possible so that it is manageable for the user. On the other hand, it should be large enough to reduce navigation within the system to a minimum. To achieve this aim, the overall functionality of the SAP System is divided into several "levels".

[Main Menu Level \[Seite 73\]](#)

[Application Level \[Seite 74\]](#)

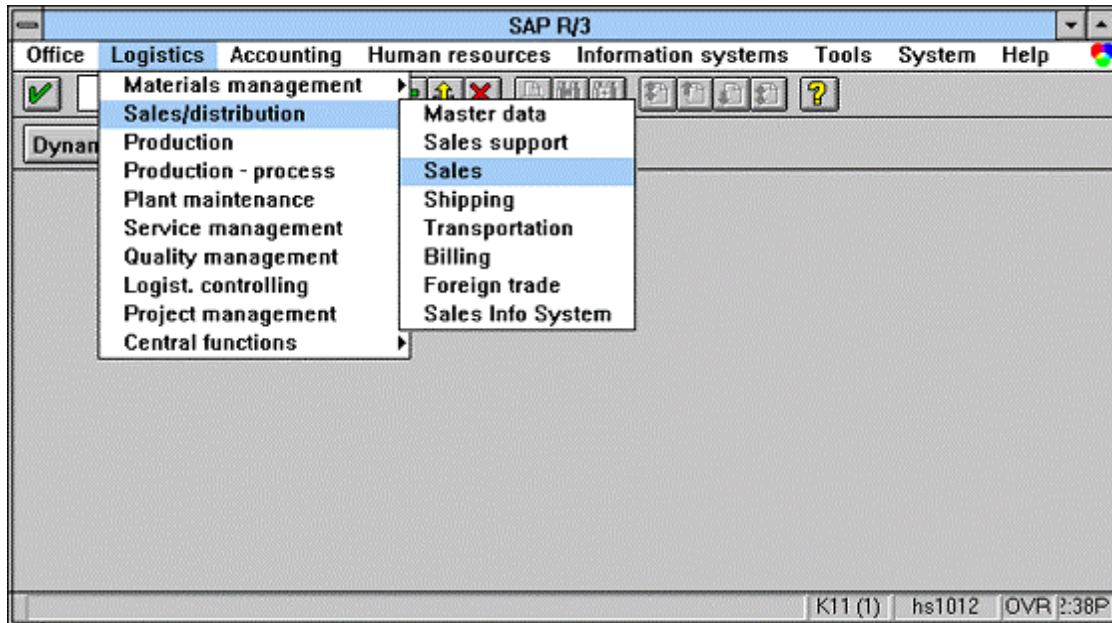
[Task Level \[Seite 75\]](#)

[Cross-References \[Seite 76\]](#)

Main Menu Level

The topmost level with the standard SAP menu displayed after logon to the standard system is called the **main menu level**. All applications offered by SAP are listed in the pull-down menus at the lowest level. The applications are grouped together in classes in the higher-level pull-down menus.

The *Master data*, *Sales* and *Shipping* applications, for example, are in the pull-down menus at the lowest level of the higher-level class *Logistics* listed in the menu bar.

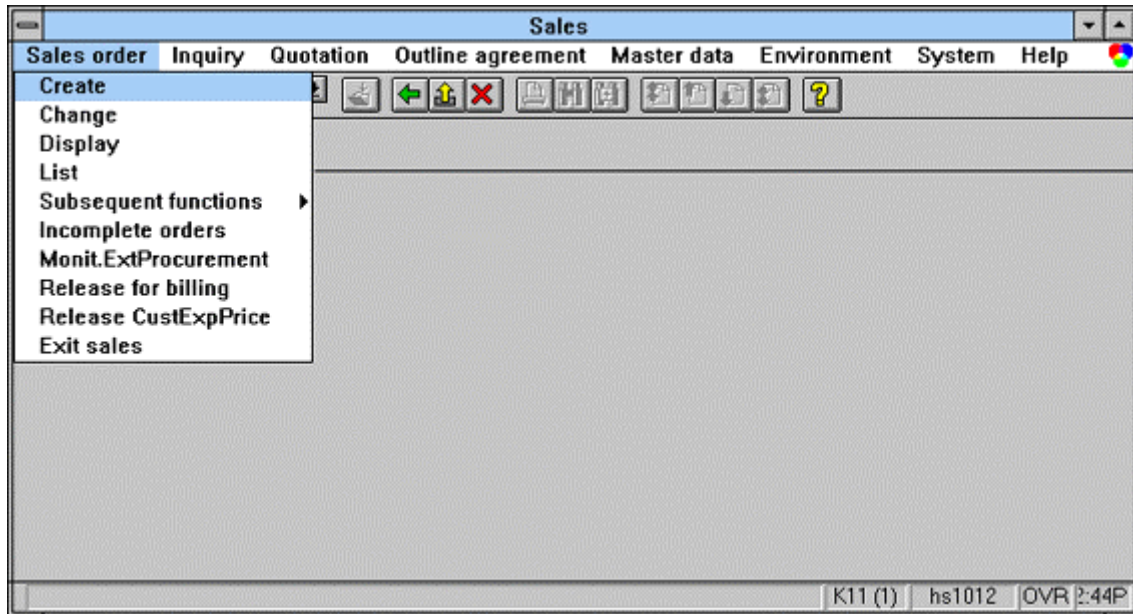


The main menu level consists of one level only. The pull-down menu of the lowest level displayed here contains the applications *Master data*, *Sales*, *Shipping* and *Billing*. *Logistics* is the topmost class to which these applications are allocated.

Application Level

Application Level

The second level of the standard system is called the **application level**. It comprises a useful number of object classes or tasks in accordance with the criteria mentioned above. A task is started by choosing a pull-down option of the lowest level, that is, an *action*. These actions are, for example, *Create*, *Change* or *Display*.



An example of a screen at application level. It is the application *Sales* from *Logistics*. Various object classes and actions can be chosen here for starting a task, for example, the object class *Sales order* and the action *Create*.

Task Level

The **task level** is the third level in the standard system and contains functions for processing a business or system task. Generally, you use a task to process a particular object of an object class - for example, *Create sales order*.

The task level is the level where object data is displayed or edited in the work area of the window. Each task generally consists of a number of task functions, such as *Save*, *Delete* etc..

The design of the menu bar is largely consistent in all windows at this level. To maintain a consistent task appearance, there are specific guidelines for the structure of the individual pull-down menus and the application toolbar as well as the allocations of the task functions to certain function keys.

The screenshot shows the 'Create Sales Order' window in SAP. The title bar is 'Create Sales Order'. The menu bar includes 'Sales document', 'Edit', 'Overview', 'Header', 'Item', 'Environment', 'System', and 'Help'. Below the menu bar is a toolbar with various icons. A row of buttons includes 'Single-line entry', 'Double-line entry', 'Business data', 'Ref.to quotation...', and 'Ref.to sales order..'. The main area contains 'Order type' set to 'OR' and 'Sales Order'. A section titled 'Organizational data' includes fields for 'Sales organization' (0001 Westcoast), 'Distribution channel' (01 DC 01), 'Division' (01 Product division 01), 'Sales office' (empty), and 'Sales group' (empty). The status bar at the bottom shows 'K11 (1)', 'hs1012', and 'OVR 2:42P'.

An example of a screen at task level. Here, the initial screen for the creation of a sales order.

Cross-References

Cross-References

- Einstieg in eine Anwendung
- [Menüleiste der Anwendungsebene \[Seite 559\]](#)
- [Wildcard Characters \[Seite 143\]](#)
- [Functions - Overview \[Seite 451\]](#)
- Menu Bar at Task Level
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Object-Oriented User Interface \[Seite 129\]](#)
- [Title Bar \[Seite 152\]](#)

Accessing Tasks

The user interface of the R/3 System is based on an object-oriented approach which has an impact on the handling of the system and the way the user starts a task.

[General Procedure \[Seite 78\]](#)

[Access via the Standard Menu \[Seite 79\]](#)

[Access via Access Characters \[Seite 80\]](#)

[Example \[Seite 81\]](#)

[Cross-References \[Seite 83\]](#)

General Procedure

Choosing the Object Class

In the last dialog step before starting the actual object processing, the user chooses a certain **object class** at application level (for example, quotations, invoices, documents, master data, plans, programs, tables, or - even more specific - GR/IR document, material master, networks, screens) (see [Levels in the SAP System \[Seite 72\]](#)). The application level with its object classes constitutes the focus of navigation for the user: From the logon screen via the main menu level the user jumps directly to the application level and returns to this screen upon completion of an application.

Choosing an Action

After choosing an object class, the user chooses the required **action** from a pull-down menu.

The object classes and actions should be geared to the actual work objectives of the user rather than reflect the logical business structure of the application area if this does not correspond with the work processes. **Processes** such as Transfers or Incoming Payments must also be possible at object class level.

Choosing a Particular Object

On the following fullscreen initial screen of the task, a particular **object** of the chosen class is called by entering a number or a name. Additionally required parameters can be entered either on the initial screen or in a separate dialog box.

Processing Individual Object Components

The actual processing of the object starts after this. The processing of the object is the sum of all the processing actions to the individual **object components**. The user chooses the required functions primarily from the menu bar.

The object-oriented user interface approach must flexibly support the various **learning stages** of the users (inexperienced user/expert user), the various **workplaces** and tasks (routine work/bulk data processing/variable information gathering), and the various **input elements** (keyboard/mouse). Therefore, the user must have various access and interaction options available to him or her.

Access via the Standard Menu

General Navigation and Syntax

1. After logging on, the user chooses the application from the **pull-down menus** presented at the main menu level. For example, Accounts payable.
2. Within this application, the user chooses the desired object class from the **menu bar** or from the **pull-down menus**.
3. From the following level of the pull-down menu, the respective **action** is chosen (object class level). If only a few actions are allocated to an object class, object class and action can be grouped together to form **one** pull-down menu option. For example, Enter goods receipt, Change master data.
4. On the task's initial screen, the object name or the **object number** is entered or chosen (task level).
5. When exiting the task, the user returns to the application level.

Choosing the Menu Options

The following options must simultaneously be available:

- Cursor positioning on the menu bar, with F10, or with the mouse pointer.
- Direct selection of menu options by single mouse click or by using the arrow keys and ENTER.
- Choosing a menu option by pressing the ALT key and an mnemonic access character (see [Access Characters \[Seite 184\]](#)) or by entering shortcuts.

Access via Access Characters

Access via Access Characters

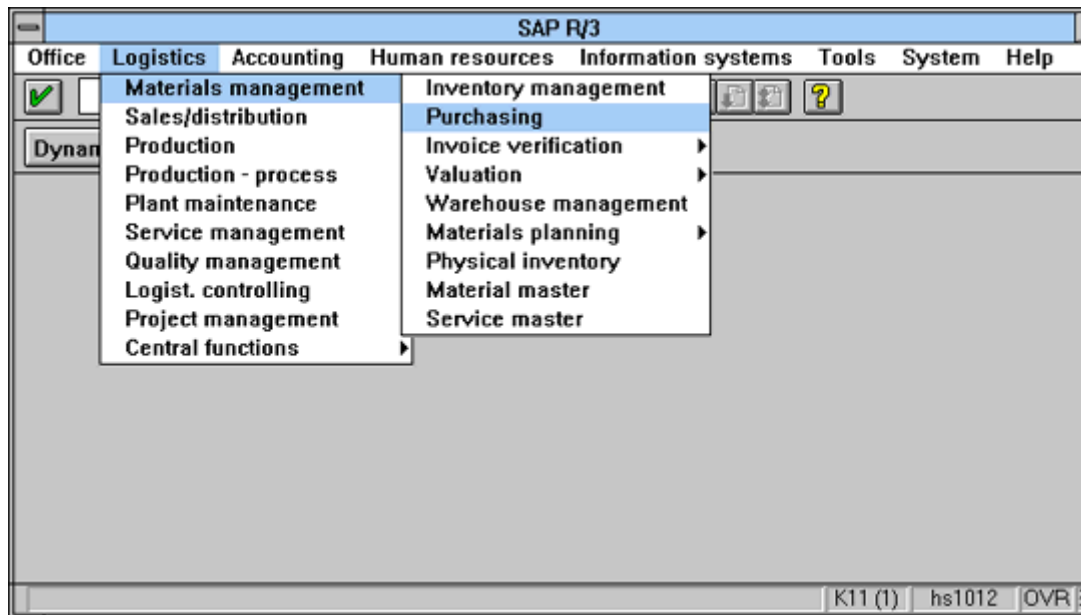
The user must be able to access the application level after logon by entering a fast path (mnemonic [Access Characters \[Seite 184\]](#)). To do so, the user enters a period and a sequence of access characters in the command field.

Transaction codes can also be supported.

Accessing the initial screen for entering the object number or the object name with an additional fast path/transaction code is also possible.

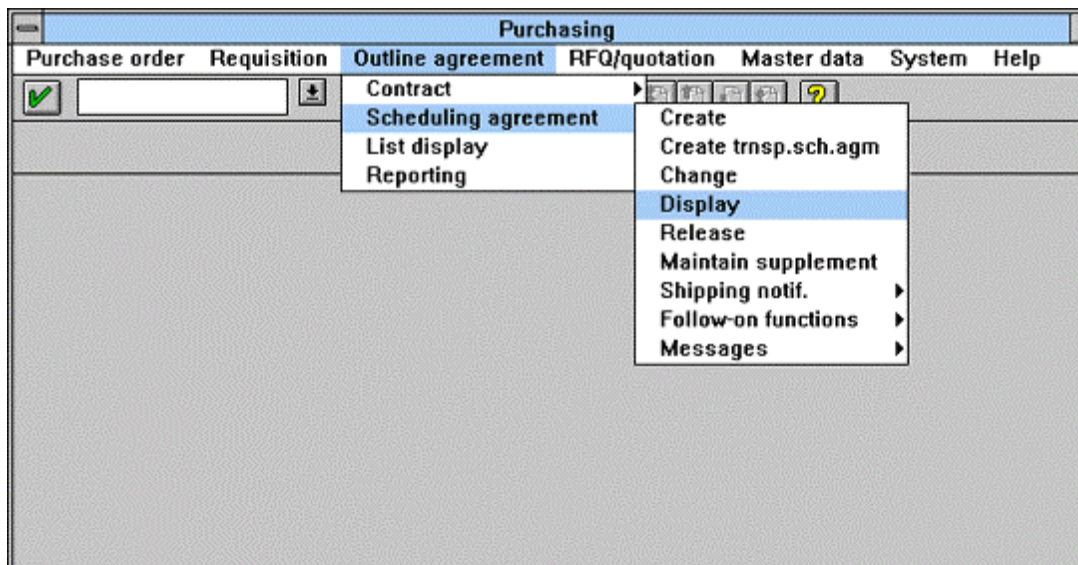
Example

1. Choosing the Application



At main menu level, *Purchasing* is chosen as a business operation area.

2. Choosing the Object Class and Actions

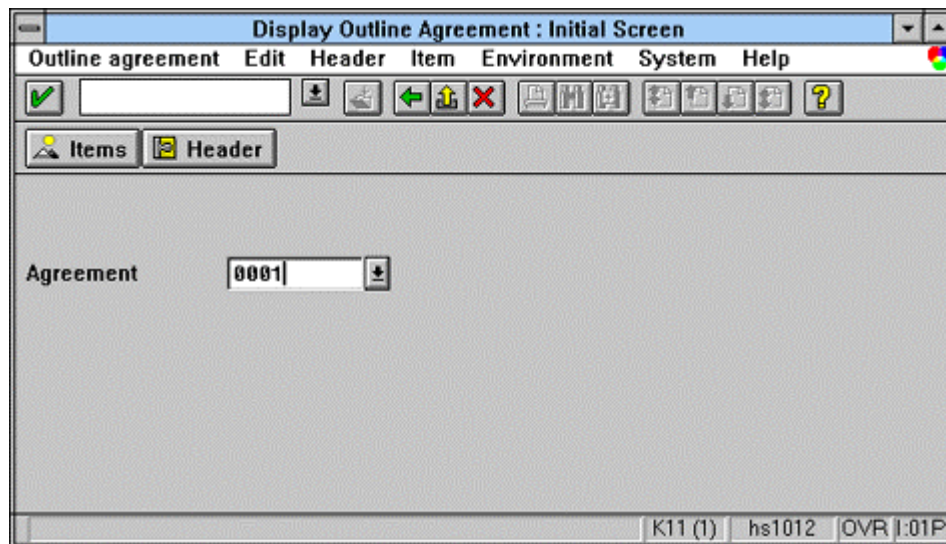


Example

At application level, the object class *Scheduling Agreement* is chosen as a form of *Outline Agreement* and the action *Display*.

3. Entering the Object Number on the Initial Screen

Follow-up screen of access option 1 (standard menu) and access option 2 (access character):



At task level, the actual object (here *Agreement*) first has to be identified on the initial screen with its number.

Cross-References

- [Access Characters \[Seite 184\]](#)
- [Command Field \[Seite 201\]](#)
- [Levels in the SAP System \[Seite 72\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Object-Oriented User Interface \[Seite 129\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Exclusive OR Selections on Initial and Selection Screens

Exclusive OR Selections on Initial and Selection Screens

When entering values into fields on initial and list selection screens, sometimes the user has to decide which of several fields he or she wants to complete. The user can only complete one of two fields (exclusive OR) as for example the fields *Cost center* and *Cost center group* in cost center accounting. In the first field, the user can specify only one cost center. If the user enters a value in the second field, a set of cost centers is considered.

This section shows possibilities how to solve this problem.

[Two Alternatives Linked With “or” \[Seite 85\]](#)

[More Than Two Alternatives with Radio Buttons \[Seite 87\]](#)

[Cross-Reference \[Seite 88\]](#)

Two Alternatives Linked With “or”

If two alternative options are linked with “or”, you can use one of the following arrangements. These suggestions apply to screens and to list selection screens.

Vertical Arrangement (Normal case)

The alternative options are arranged one below the other. Separate the alternatives by one line which includes the word “or”, indented by three characters. The word “or” is to appear in lower case and in normal display (not intensified).




```

Cost center      _____
    or
Cost center group _____
    
```

Horizontal Arrangement

If vertical arrangement is not possible due to space reasons, the following horizontal arrangement is permissible:




```

Cost center      _____ or Cost center group _____
    
```

Variant with “from-to” Field

If an alternative includes a field which allows the user to enter a range, the arrangement should look as follows:



```

Cost center      _____ to _____
    or
Cost center group _____
    
```



Example with Check Text

Screens with check texts also allow the following display:

```

Cost center      _____ <Check text>
                                to _____ <Check text>
or      Cost center group _____ <Check text>
    
```

More Than One Field For Each “OR” Alternative

If one alternative belongs to several fields, they have to be presented one below the other in a vertical arrangement without a blank line separator. Fields which do not refer to the “or” alternatives have to be separated by a blank line.

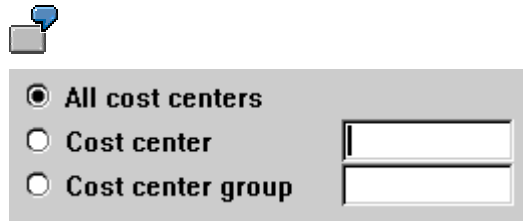
Two Alternatives Linked With “or”

Cost center _____ to _____
Business area _____
or
Cost center group _____

More Than Two Alternatives with Radio Buttons

If you use more than two alternatives, in particular if a field of the type “all xy” is included, radio buttons should precede the fields presented in a vertical arrangement. The word “or” is of course not required then.

Three And More Alternatives



The screenshot shows a SAP form with three radio button options: "All cost centers" (selected), "Cost center", and "Cost center group". To the right of these options are two empty input fields. A small blue speech bubble icon is visible above the first input field.



If the user enters accidentally data in several fields, the field for which the radio button is set is active. The field which is expected to be selected most often is always to be placed at the beginning and/or be ready for input.

Instead of a field name/input field arrangement, you can also use the checkbox/field name arrangement. The checkbox is to be positioned to the right of the field name.

Technical Requirements to the Basis Department

Besides the “graphical element/name” arrangement, it should be possible to define a composed object which includes several fields. This makes only the field(s) ready for input, for which the user has selected the radio button(s). If the user clicks on the input field, the pushbutton is switched and the field is ready for input.

Cross-Reference

Cross-Reference

- [Searching in Structured Data \[Seite 142\]](#)

Effects on Perception

A number of psychological principles (the Gestalt concept of perception) can be employed to improve the perception of information.

This section explains the four most important principles with regard to how we perceive.

[The Effect of Proximity \[Seite 90\]](#)

[The Effect of Similarity \[Seite 91\]](#)

[The Effect of Closure \[Seite 92\]](#)

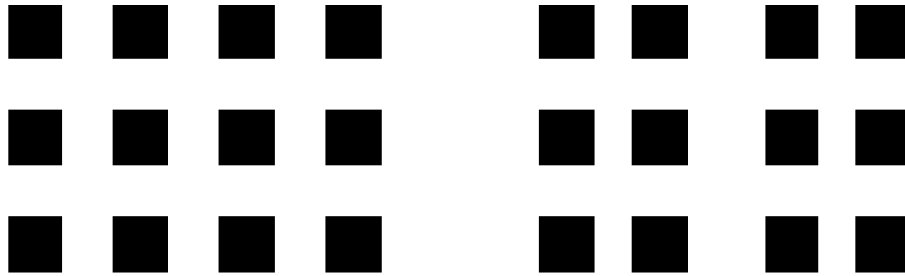
[The Effect of Continuity \[Seite 93\]](#)

[Cross-Reference \[Seite 94\]](#)

The Effect of Proximity

The Effect of Proximity

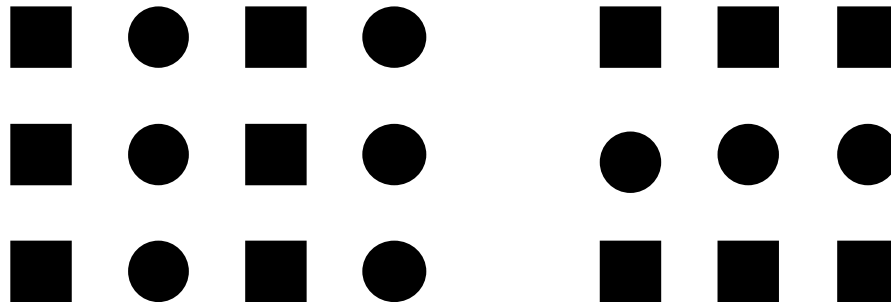
Items that are close together tend to be 'grouped' in our perception.



The effect of proximity on perception

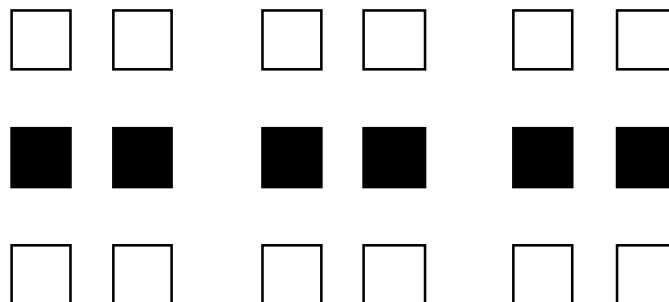
The Effect of Similarity

Items of the same size, shape, or quality are more likely to be viewed as a group or pattern than dissimilar elements.



The effect of similarity on perception

If we look at the following figure, we see that conflicts can arise when the effects of proximity and similarity interfere with one another or even cancel each other out.

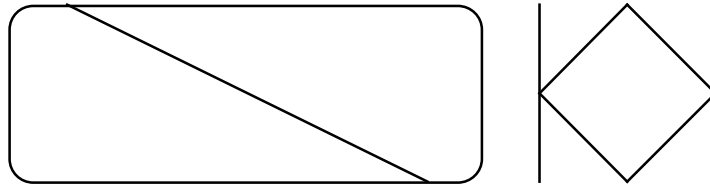


Conflict between similarity and proximity

The Effect of Closure

The Effect of Closure

Lines which enclose areas are perceived as units, in contrast lines which are not closed.



The effect of closure on perception

The Effect of Continuity

Items displayed in a "good" arrangement are perceived as a unit. This applies particularly to alignments on the template.



Field name-1	1231948
Field name-2	434234
Field name-3	744
Field name-4	572457

Alignment

The effect of continuity on perception

Cross-Reference

Cross-Reference

- [Fields - Positioning \[Seite 253\]](#)

Selection Methods - Overview

This section describes the methods currently used in the R/3 System for selecting objects in table structures.



This section deals with selection in tables which use the step loop technique. For new developments use the [Table Control \[Seite 340\]](#) technique (**exception:** multiple-line tables), which supports selection to a great extent automatically. Otherwise the guidelines described in this section apply.

[Terms \[Seite 96\]](#)

[General Guidelines \[Seite 97\]](#)

[Types of Selection \[Seite 99\]](#)

[SAP Matrix \[Seite 103\]](#)

[Special Character '*' in the Selection Field \[Seite 101\]](#)

[Technical Notes \[Seite 104\]](#)

[Cross-References \[Seite 105\]](#)

Terms

Terms

Objects

The objects of a table structure are the screen elements which the user can select and edit. These can be areas or contents of varying size, depending on the table structure and technical possibilities. Typical objects of a table structure are individual cells, entire lines or an entire column. "A line" here means an entire logical line - that is, both physical lines for tables with two lines per record.

Select

To **select** an object means to emphasize an object to be acted upon or an action to be performed. The object will be affected by any action the user initiates subsequently, e.g., the user chooses the [Delete \[Seite 515\]](#) function.

The Windows File Manager is a good example: If the user clicks on a file, the file is selected. To perform a function on the selected file (e.g. to delete the file), the user has to choose a menu option (here, *Delete*) (see [Object-Oriented User Interface \[Seite 129\]](#)).

Deselect

The term used in the R/3 System for cancelling a selection is **deselect**.

Types of Selection

Depending on the task to be performed, the user can use a specific type of selection from a table structure. Selection may be classified as single or multiple. If only **one** item can meaningfully be selected, we talk about **single selection**. If **several** items can be selected simultaneously, we talk about **multiple selection**.

General Guidelines

Explicit Selection and Deselection

With the Keyboard

The user positions the active control indicator on the desired object and presses the Select key. The selection state of the entire object changes: If the object is not selected, it is selected now; if it is selected, it is now deselected (for information on the Select key, please refer to the "Technical notes").

With the Mouse

The user positions the mouse pointer on an object and presses the mouse button. The selection state of the entire object then changes (see Technical Notes).

For more detailed information, please refer to [Select/Deselect \(Function\) \[Seite 516\]](#).

Automatic Selection

Sometimes it is more efficient if the the object is selected when the user positions the cursor on it. With character-specific user interfaces, the selection is indicated merely by the cursor position. On graphical user interfaces, the mouse-based automatic selection also has to be linked with the display of the selection state (see Technical Notes).

With the Keyboard

The user positions the cursor on the required object. The object is thereby selected. (Note: Until the selection is displayed, the object is selected merely by the position of the cursor. The user can now directly perform an action on an object (by choosing a menu option or pressing the function key or `ENTER`).

With the Mouse

The user positions the pointer on an object and presses the mouse button. The object is thereby selected. The user can now directly perform an action on the selected object (by choosing a menu option or pressing the function key or `ENTER`), or select further objects, if multiple selection is possible.



Automatic mouse selection is identical to explicit mouse selection.

Priority between the Principles

Explicit selection is the basic selection method. Explicit selection must always be supported for both the keyboard and the mouse. Explicit selection can be replaced by automatic selection, if appropriate.

If an object has already been selected and the cursor is on another object of the same table structure, the action initiated by the user is only performed on the selected object. This means that cursor-based automatic selection in a table structure is only possible, if the user has not explicitly selected an object.

General Guidelines

If the user positions the mouse pointer on an explicitly selected object and presses the mouse button, the object in question is deselected and not automatically selected. If two table structures are available on a template, however, you can use automatic selection in the second table structure, although explicit selection was performed in the first structure. The developer has to define if the action the user initiates applies to all of the selected objects or only to one. The developer must decide how the action is to be implemented and name the action correspondingly.

Indicating Selection

Selected objects have to be visually distinguishable from non-selected objects. In the R/3 System, selected objects currently must have the attribute "highlighted".

Temporary regulation: As long as selection with a single mouse click is not supported locally, you can position the active control indicator to indicate single selection or automatic selection.

In future, you will be able to use the "selected" attribute for program-driven selection. This allows to indicate an object as selected at least after a host interrupt. The specific appearance of the selection state will be determined locally by the GUI server.

Positioning the Cursor

If only the table structure is displayed on a screen, or if it is the dominant element of the screen to be processed, the system positions the cursor automatically on the first object in the table structure when the screen is first displayed.

If you have branched from a table structure to the details of a line and then return to the table structure, the cursor is positioned on the object of which you have last viewed the details.

Types of Selection

Selection may be classified as **single** or **multiple**.

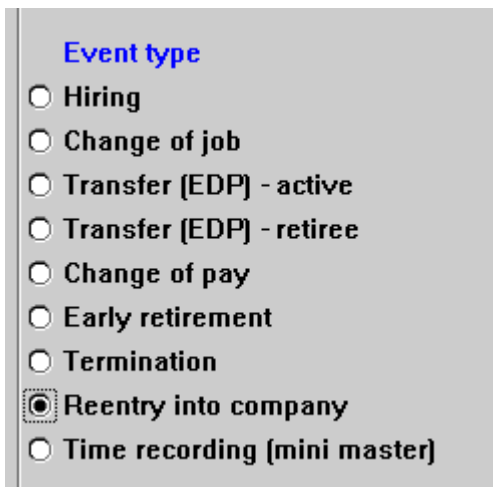
Single Selection

Definition and Use

Single selection allows the user to select a single object in the table structure. Selecting an object automatically deselects a previously selected object. The table structure is not ready for input. Objects which can be selected are entire logical lines.

Appearance

For one-column or two-column tables, you can use a radio button to perform the single selection. For multiple column tables, do not use a radio button until further notice.



The screenshot shows a table with a single column titled "Event type". It contains a list of options, each preceded by a radio button. The option "Reentry into company" is selected, indicated by a filled radio button and a dashed rectangular border around the text. The other options are "Hiring", "Change of job", "Transfer (EDP) - active", "Transfer (EDP) - retiree", "Change of pay", "Early retirement", "Termination", and "Time recording (mini master)".

Table with single selection and radio buttons

Selection Methods

- Explicit selection and deselection of an object (see above)
 - The user can click on a radio button, if available. After the user has pressed `ENTER`, the task must display the object as selected.
- Automatic selection (see above)

Technical restriction: For step loops, the system does not set any default radio buttons. Furthermore, the task must set the radio buttons itself when the user uses the scroll function.

Multiple Selection

Definition and Use

The user selects any number of objects (0 to n) in the table structure. The selection status of previously selected objects does not change when the user selects an object. Only entire logical lines can be selected, not fields or columns.

Types of Selection

Appearance

If several lines can be selected simultaneously in a table structure (multiple selection), the selection fields are to be displayed as checkboxes.

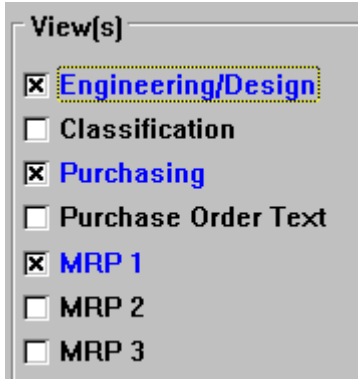


Table with multiple selection and checkboxes

In two-line table structures, checkboxes can only appear at the beginning of every logical line.

If a line is selected, it should be highlighted by the task. To achieve immediate selection highlighting, the F9 functionality should be supported in addition to the mouse click.

Exception: If the table structure is primarily used for entering bulk data, the entry mode does not require a checkbox. To select lines, the user then must enter the display or change mode in which the table structure must have a checkbox in front of every line.

Selection Methods

Individual Objects

- Explicit selection and deselection of an object (see above)
 - If available, the user can click on the checkboxes. After the user has pressed `ENTER`, the task must display the objects as being selected.
- Automatic selection (see above)

Object Groups

- Selecting and deselecting **all objects** by choosing the function [Select all \[Seite 473\]](#) or Deselect all.
- Furthermore, a function should be displayed to the user for selecting **any blocks** (see [Select Block \[Seite 494\]](#)).

These functions should be available via the [Edit \[Seite 565\]](#) menu.

Special Character '*' in the Selection Field

Previous Situation

Previously, the task program could flag the objects edited or viewed in detail by the user, by entering '*' in the selection field.

The user could not manually enter the '*'. Other special characters for indicating a specific status were also not allowed.

Graphical Checkboxes

The implementation of graphical checkboxes has changed the situation. For technical reasons, you can currently not use asterisks in graphical checkboxes to indicate the "Edited" status. A separate column must be implemented in front of the checkbox. The one-character field receives the flag FixA. Otherwise, the guidelines listed below still apply.

	Heading	Heading	Heading
* <input checked="" type="checkbox"/>			<input type="checkbox"/>
<input type="checkbox"/>			<input type="checkbox"/>
* <input type="checkbox"/>			<input type="checkbox"/>
<input type="checkbox"/>			<input type="checkbox"/>

Entry 3 of 15

Edit Flag

The "Edit" flag is displayed in a separate column in front of the checkboxes



You see a table structure with items of an order document. You select five items and enter the account assignment to three items in 'fullscreen'. After entering the third item, you return to the table structure. The first three items show an '*' in the "Edited" column, the other two a blank.

When using an '*', be aware of the following differences:

Repeated Display of '*' If Only One Action is Possible

If only one action can be performed on the objects of a table structure (for example "Choose"), once set, '*' can remain set even though the user toggles several times between the table structure and the detailed view.

Special Character '*' in the Selection Field



You are working with an order document and want to copy items from another order document into your current document. You therefore select and copy items in a table structure which shows you the items of the "old" reference document. Asterisks '*' are set as reminders in the fields of the table structure to indicate which items of the "old" document you have already copied.

You continue working with the "new" document and want to copy further items. If you display the table structure with the items of the "old" document, the asterisks '*' are still there. Even if you access the table structure of the "old" document five times, the '*' do not disappear; instead, new ones are added if you copy further items from the "old" document.

Once-Only Display of '*' If Several Actions are Possible

If several actions are possible on the objects of the table structure (for example "Maintain the account assignments to this item" and "Maintain the partners to this item"), you only see the '*' when you return to the table structure for the first time. When you return to the table structure a second time, the asterisks '*' are no longer there.



You see a table structure with items of an order document. You select three items and enter the account assignment to them. When you are finished, you return to the table structure and see a '*' as a reminder of the three items you have processed.

You now enter the partner data for four other items. When you return to the table structure again, you see the '*' only for the items you have just processed. The '*' from the first processing with the account assignments are no longer there.



You can optionally implement a "Select again" function to reselect all lines with '*'.

SAP Matrix

Definition and Use

"SAP matrix" in this context means a table structure ready for input containing fields, lines and columns as selectable objects. The user can select as many lines, columns and fields as desired.

Appearance

Until further notice, an SAP matrix is displayed as a table structure with multiple selection (see [Selection Tables and Lists \[Seite 362\]](#)).

Selection Methods

Fields:

explicit selection and deselection (see above). The selected field can then be processed (that is, edited) with any function.

Lines:

as with multiple selection (see above). The cursor, however, has to be positioned on the checkbox in front of the lines.

Columns:

an entire column is selected and deselected by explicitly selecting and deselecting the column header (see above).

Checkboxes can select lines or fields, depending on the layout.

Planned Enhancement

The checkboxes in front of the lines and the headers will become pushbuttons, as in spreadsheets. The selection of several lines occurs sequentially by means of "[Select block \[Seite 494\]](#)" or by dragging with the mouse. Several columns or fields may be selectable in the same way. The task developer must define the valid selection possibilities in the matrix in accordance with the user's needs.

Technical Notes

Explicit Selection and Deselection

With the Keyboard

Temporary solution:

Until further notice, the Select key is the function key F9 so that a host interrupt is generated. The Basis group is investigating the possibility of controlling select functionality locally. In this case, the space bar defined by the various platforms should replace the F9 key.

With the Mouse

Temporary solution:

Currently it is not possible to select objects with a single mouse click. However, the Basis group will shortly provide this functionality. Until then, the cursor will be set with the mouse without display of the selection status. This is automatic selection (see above).

Automatic Selection

As yet, the "select" state is not visible on the screen during automatic selection with the cursor.

Cross-References

- Ankreuzfelder (Bedienelement)
- [Einfachauswahllisten \[Seite 365\]](#)
- [Mehrfachauswahllisten \[Seite 363\]](#)
- [Checkboxes \(Control Element\) \[Seite 176\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- Multiple-selection list
- [Object-Oriented User Interface \[Seite 129\]](#)
- [Radio Buttons \(Control Element\) \[Seite 193\]](#)
- [Select All, Deselect All \(Functions\) \[Seite 473\]](#)
- [Select Block \(Function\) \[Seite 494\]](#)
- [Select/Deselect \(Function\) \[Seite 516\]](#)
- Single-selection list
- [Selection Tables and Lists \[Seite 362\]](#)
- [Table Control \[Seite 340\]](#)

Navigation Between Screens

Navigation Between Screens

This section describes typical screens in the SAP System and the navigation options which should be offered to the user on these screens.

If possible, the following information is displayed on every screen:

- Function
- Navigation options at higher hierarchy levels
- Navigation options at the same hierarchy levels
- Navigation options at lower hierarchy levels
- Navigation options on the current screen

[The Screen Calling the Task \[Seite 107\]](#)

[The Initial Screen \[Seite 109\]](#)

[The Object Component List Screen \[Seite 112\]](#)

[The Object Component Screen \[Seite 117\]](#)

[The Sub-Component of Object Screen \[Seite 125\]](#)

[Cross-References \[Seite 128\]](#)

The Screen Calling the Task

Function

On this screen, the user chooses a class of objects and then specifies the function to be executed. The user can do this in one of the following ways:

- choosing an object class and a function from a menu



The RF application menu Accounts Receivable contains the items *Create master records* or *Display document*. The Logistics application menu (Materials management - Purchasing) contains the items *Change purchase order* or *Display RFQ/quotation*.

- entering a fast path which contains an object class and a function,



Many tasks are designed this way, for example, "Maintain RP master data", "Create accounting document" or "Open ABAP/4 program".

- choosing a function which calls another task within a task. In the other task, an object class and function are also chosen.



From the user's point of view, the process described here is actually the same as the one described first. It is listed separately here because the technical background is different.

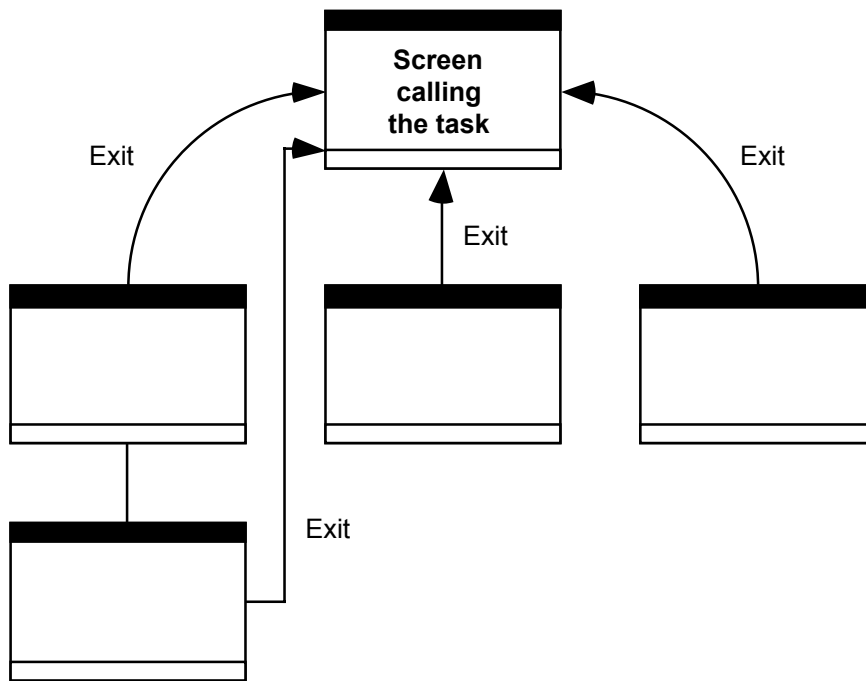
Navigation Options at Lower Hierarchy Levels

As described above, the user either chooses a pull-down menu item or a function key or enters a fast path and chooses [Continue \(Function\) \[Seite 540\]](#).

Navigation Options to "the Screen Calling the Task"

In addition to the branching possible within "the screen calling the task", the *Exit* function always branches to "the screen calling the task" (providing this is technically possible and meaningful).

The Screen Calling the Task



Exit navigates to "the screen calling the task", if possible.

The Initial Screen

[Zurück \[Seite 481\]](#)

[Zurück \[Seite 481\]](#)

[Zurück \[Seite 481\]](#)

[Weiter bzw. OK \[Seite 540\]](#)

[Bearbeiten \[Seite 529\]](#)

[Einzelbearbeitung \[Seite 529\]](#)

[Sammelbearbeitung \[Seite 529\]](#)

[Erfassen \[Seite 503\]](#)

[Einzelerfassung \[Seite 503\]](#)

Function

On this screen, the user generally chooses a particular object which he wants to subsequently process. The user may also specify processing control, enter values for the Copy from function or select views for later processing.



For creating a vendor in RF, for example, the company code and the account group have to be entered; for changing a human resource master record in RP, the personnel number and the info type, i.e. a particular view of the human resource master record, have to be entered. In ABAP/4, the program name has to be entered.

Navigation Options at Higher Hierarchy Levels

With the "Exit" function, the user returns to "the screen calling the task", that is, exits the processing of objects in an object class. The functions [Back \[Seite 544\]](#) and [Cancel \[Seite 470\]](#) have the same effect.



After choosing the menu option *Create accounts receivable*, the system displays the initial screen. By pressing [Exit or End \[Seite 492\]](#) or Back, the user exactly returns to where he came from, i.e. generally to the application level or the main menu level (depending on the call).



Both functions should be supported. [Exit or End \[Seite 492\]](#) is an absolute branch which always branches to "the screen calling the task" whereas Back means a relative branch to the next higher hierarchy level. The Exit function is to be displayed in the pushbutton layout on the initial screen.

The Initial Screen

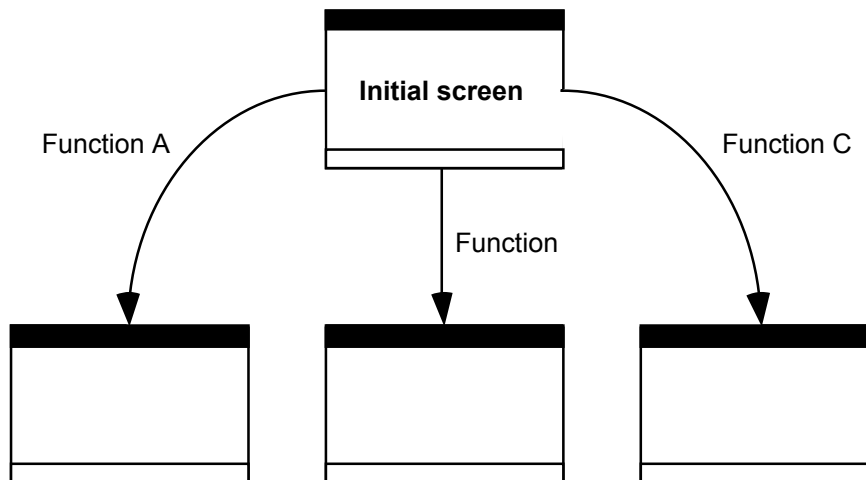
Navigation Options at the Same Hierarchy Level

There are no screens at the same hierarchy level.

Navigation Options at Lower Hierarchy Levels

- **Object components that can be freely defined**

If the object components the user actually wants to work with are contents-related groups of data and can be defined without restrictions (see below), the user can freely define navigation options for selecting a group.



Navigation options from the initial screen

Branches that cannot be carried out because they are object-dependent result in an error message.

"Continue" can carry out a default branch here.



In the R/3 SAPoffice task, there is one function to get to the screen of the inbox folder, another function to get to the outbox folder, another function to display the private folders and yet another function to display the shared folders.

If too many different views are possible, the navigation options which are used less frequently should only be selectable via pull-down menus.

- **Predefined navigation sequence**

If the next screen is predefined by the software developer, for example, a list of the object components (see below) or the first screen of a sequence of screens, the user can get there by using the [Continue \[Seite 540\]](#) function.

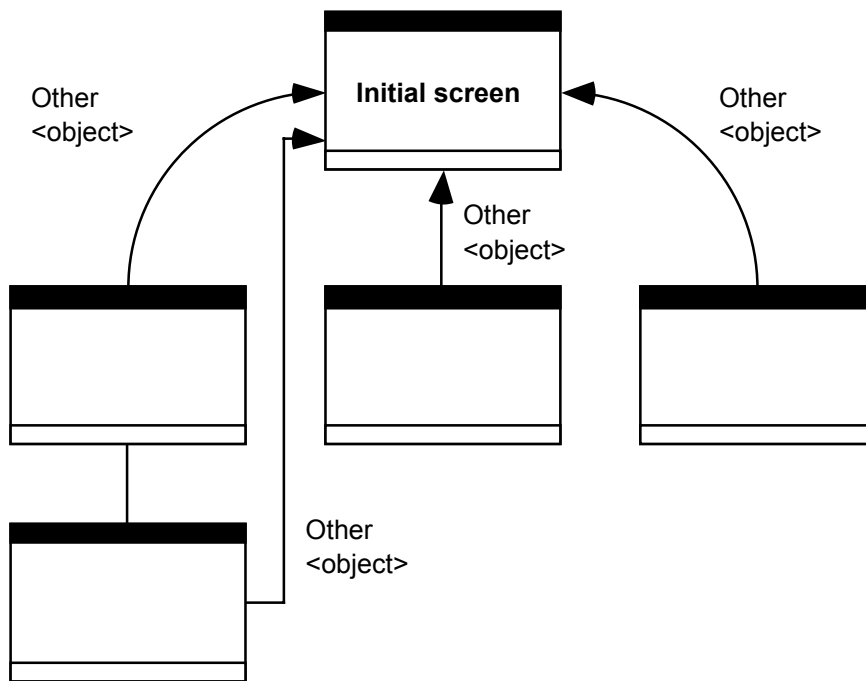
Please note, however, that predefined screen sequences should be avoided in R/3, wherever possible.



After entering the account number, company code and account group in the R/2 System under *Create vendor master records*, you get to the first object component screen which contains the basic supplier data by pressing [Continue \[Seite 540\]](#).

Navigation Options to the Initial Screen

In addition to the relative navigation options that are possible to the initial screen, the function [Other <object> \[Seite 474\]](#) always returns to the initial screen, providing this is technically possible and meaningful.



If possible, *Other <object>* navigates to the initial screen.

In addition to the function [Other <object> \[Seite 474\]](#), the standard functions [Create \[Seite 479\]](#), [Display \[Seite 481\]](#), Change, Edit, Single Processing, Collective Processing, [Enter, Single Entry, Collective Entry \[Seite 503\]](#), also always navigate to the initial screen. This is not pointed out separately in the following.

The Object Component List Screen

[Zurück \[Seite 481\]](#)

[Voriges <Teilobjekt> \[Seite 515\]](#)

[Nächstes <Teilobjekt> \[Seite 515\]](#)

Function

- **Simple lists**

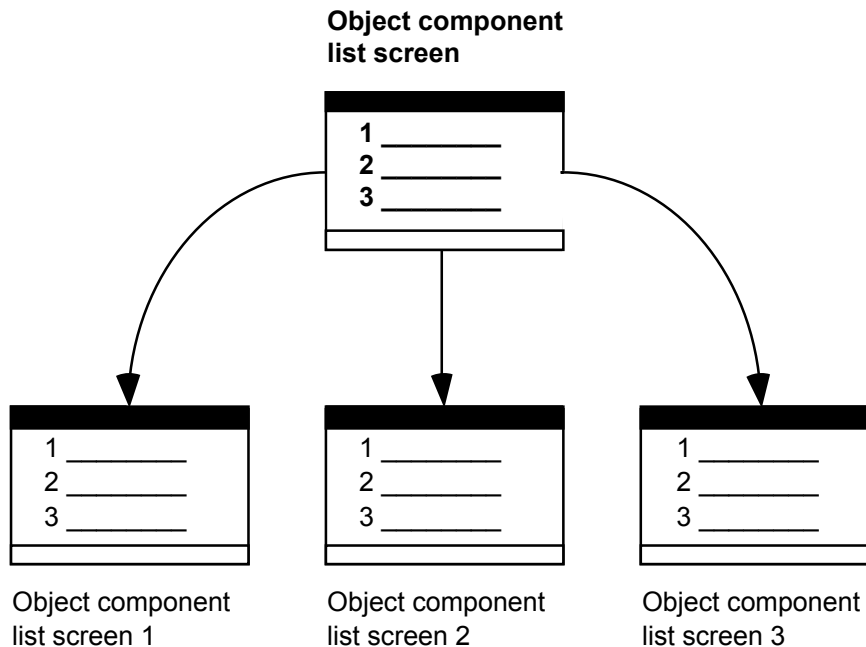
In some tasks, the user is shown a list of object components for information and navigation purposes (for example, on the screen following the initial screen).



For a document in RF, the user can display a document overview listings all line items. In RP, an overview containing all info-type records can be called for a human resource master record and infotype.

- **Lists within lists**

By means of a function, it is also possible to get from one object component list screen to other object component list screens on the same hierarchy level.



Navigation options on the object component list screen, to other object component list screens

The Object Component List Screen

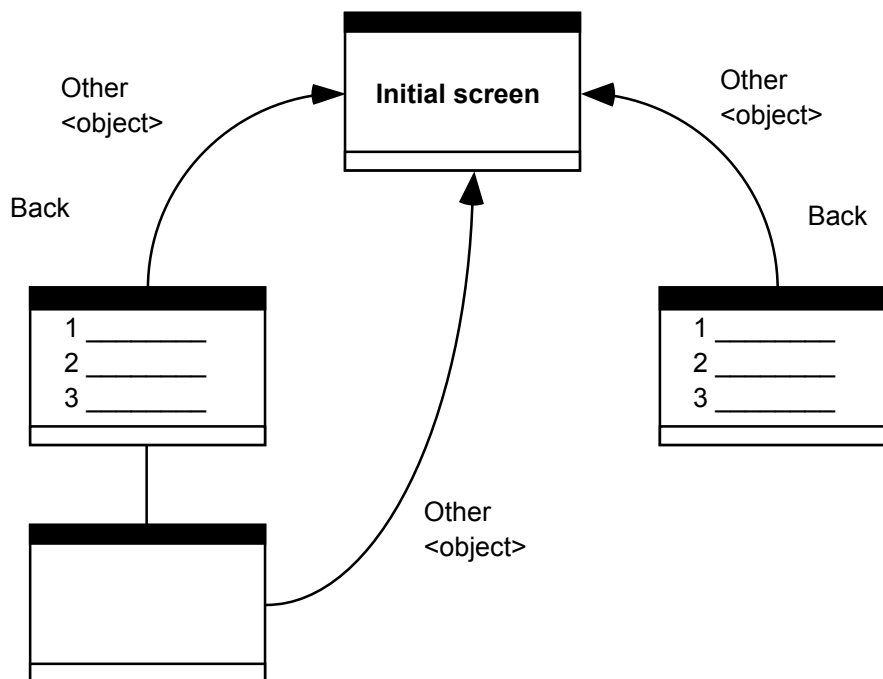


First a list of vendors is displayed. After choosing a vendor, the purchase orders with this vendor are displayed in a list. Now the user could choose a purchase order from the list.

Navigation Options at Higher Hierarchy Levels

- **To the initial screen**

The user reaches the initial screen with the function [Other <object>](#) [Seite 474]. If the object component list screen was directly displayed from the initial screen, he or she also returns to the initial screen when pressing Back (Function). With these functions, the user leaves the processing of the current object.

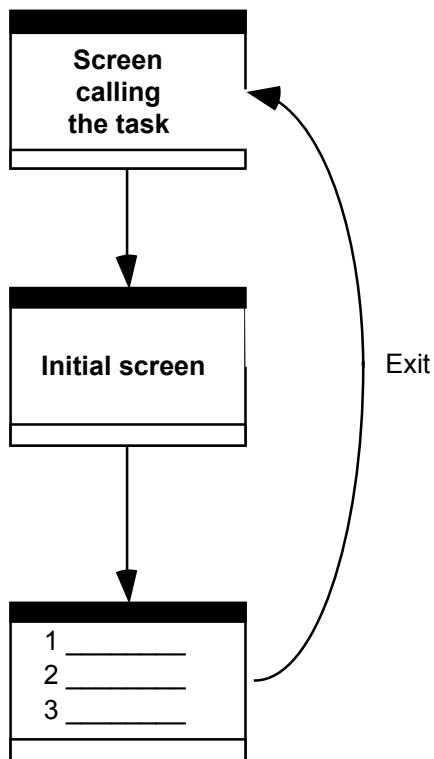


Navigation options on the object component list screen to return to the initial screen

- **On "the screen calling the task"**

The user reaches "the screen calling the task" using the function [Exit or End <application>](#) [Seite 492]; this way he exits the editing of all objects in object class.

The Object Component List Screen



Navigation options on the object component list screen to return to "the screen calling the task"

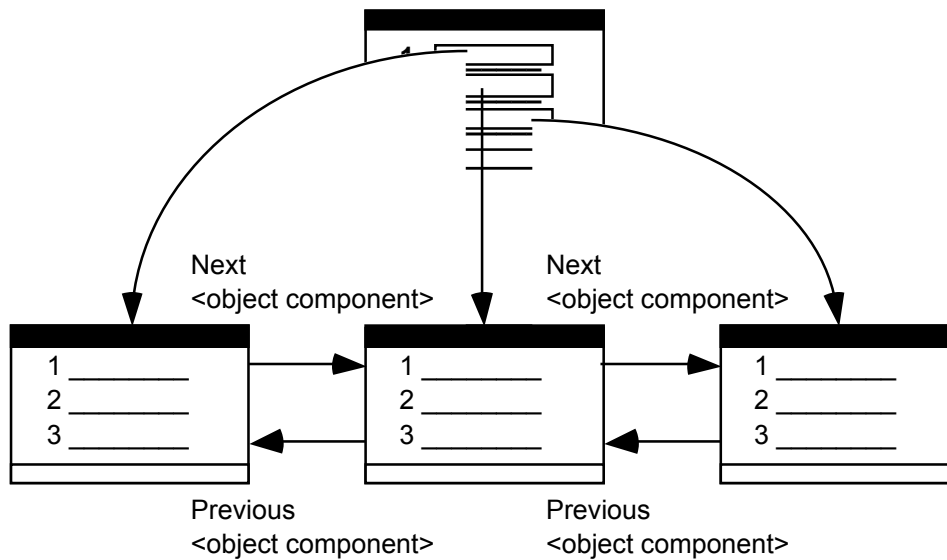


The user has chosen *Display document* and entered a document number. The document overview containing all line items is displayed. With [Exit \[Seite 492\]](#), the user exits the task *Display document* completely to perform a different task, for example, creating vendor data.

Navigation Options at the Same Hierarchy Levels

On list screens which are themselves elements of another, higher-level list, you get to the previous list using the function Previous <object component> and to the next list using the function Next <object component>.

The Object Component List Screen

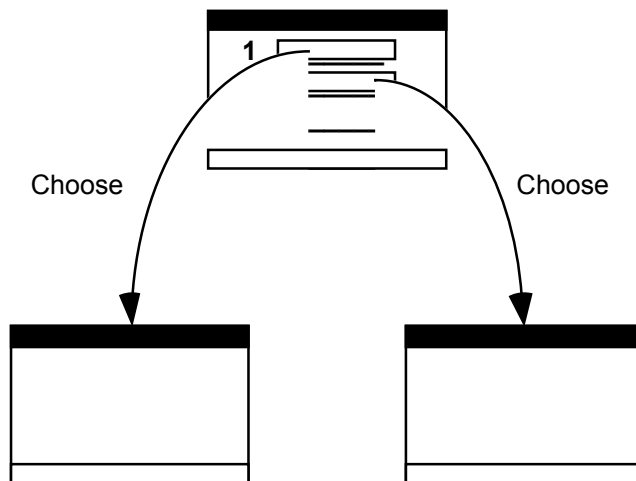


Navigation options between object component list screens on the same hierarchy level

Navigation Options at Lower Hierarchy Levels

- If only one view is possible

If there is only one screen for every line of the list, the user can choose a line by positioning the cursor on the required line and by choosing the function [Choose \[Seite 487\]](#).



Navigation options for only one view

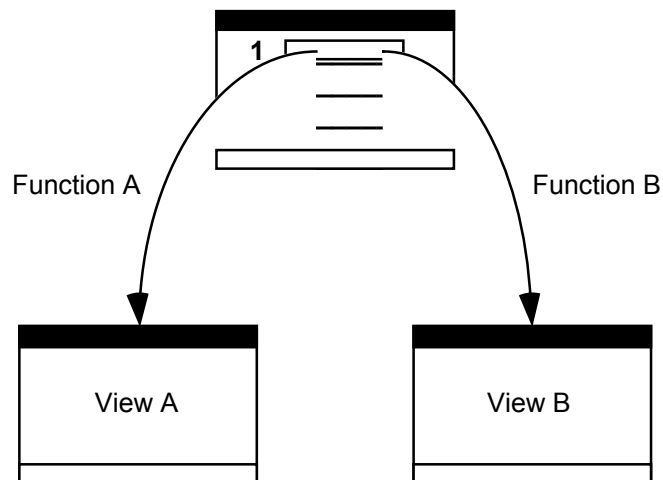


For the human resource master data, there is an overview of all address data of a human resource master record. With [Choose \[Seite 487\]](#), the user can display the details of the address data.

The Object Component List Screen

- **If several views are possible**

If several views are possible for every line, the user chooses a line by positioning the cursor on the required line. The user then chooses the view by activating a corresponding function.



Navigation options for several views



In a vendor list, you can display e.g. the address data per list entry with one function, the sales with another function, and the bank data with a third function.



Setting of the F2 key

The most frequent or most important function should be assigned to the F2 button for a fast selection with the mouse.

The Object Component Screen

[Zurück \[Seite 481\]](#)

[Anderes Objekt \[Seite 474\]](#)

[Merken \[Seite 540\]](#)

[Voriges <Teilobjekt> \[Seite 515\]](#)

[Nächstes <Teilobjekt> \[Seite 515\]](#)

[Voriges <Teilobjekt>/Nächstes <Teilobjekt> \[Seite 515\]](#)

[Voriges <Teilobjekt>/Nächstes <Teilobjekt> \[Seite 515\]](#)

Function

You can distinguish different types of object component screens.

- **Freely selectable components**

The data for an object are sometimes grouped in contents-related groups which perhaps may even be maintained and stored separately.



A program in ABAP/4 is divided into the attributes, the program source, the text elements and the documentation. Every object component has its own screen and can be maintained and stored separately.

- **Accessing the object component via an object component list screen**

An object sometimes consists of object components which are identical in structure. In these cases, the object components are often displayed in a list, from which an object component can be chosen.



An accounting document has different items which all have the same structure. An info-type human resource master record has different lines on the time axis which all have the same structure.

- **Object component as part of a "predefined sequence"**

Sometimes not all data can be included on one screen and therefore have to be distributed over several templates which are sequentially linked together.



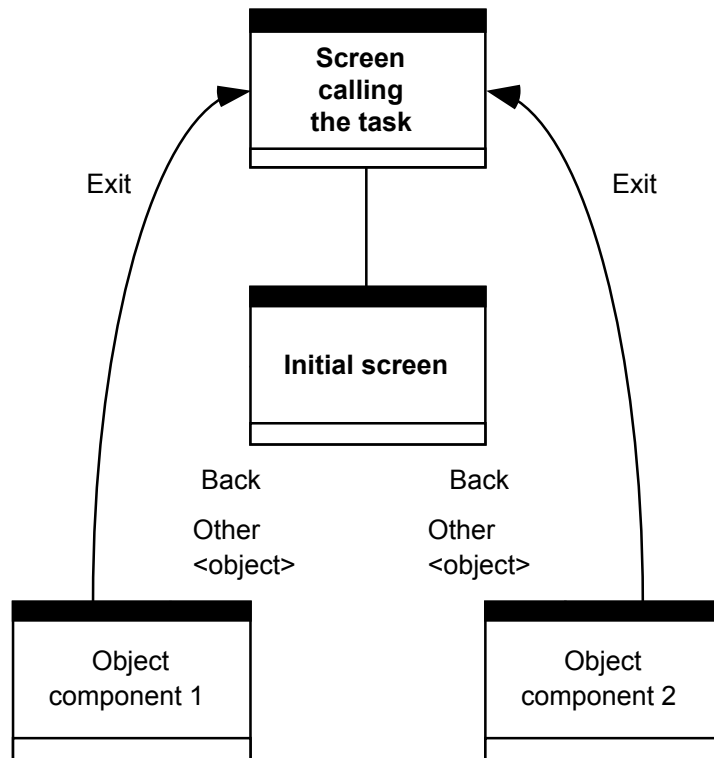
In the RF vendor master record in the R/2 System, the screens of a record are linked together: The sequence is predefined and the record can only be posted after all screens have been processed (they are not "actual" object components but the screens are treated as such).

The Object Component Screen

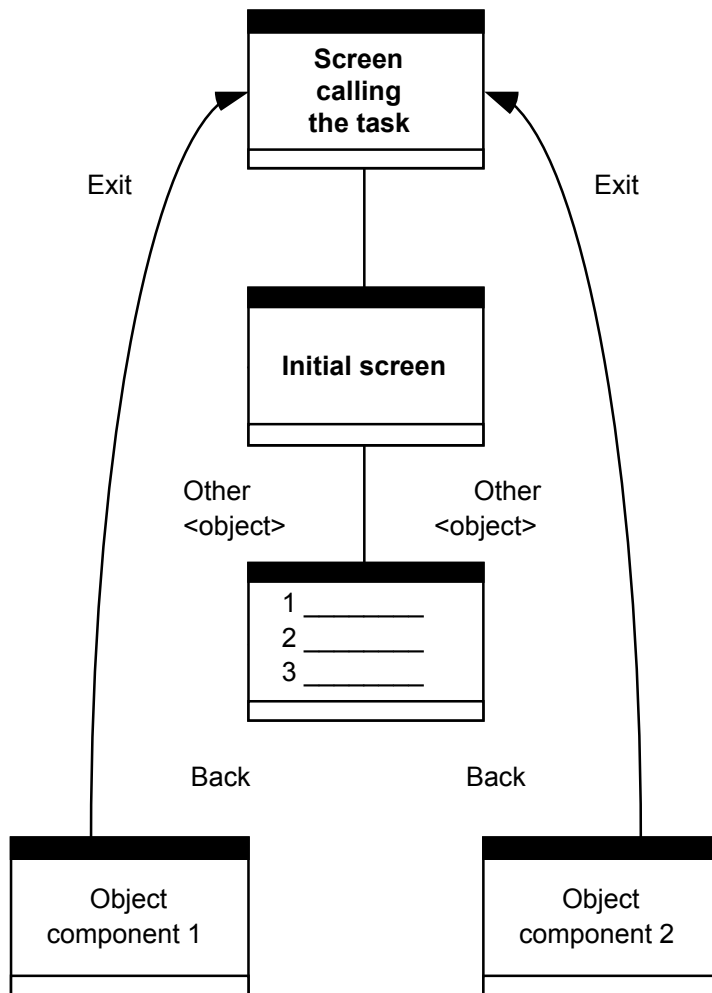
Navigation Options at Higher Hierarchy Levels

- Standard navigation

With the Back function, you get to the screen which is exactly one level higher in the hierarchy, for example, the object component list screen or the initial screen (depending on what is available or where the user came from). With the [Other <object> \[Seite 476\]](#) function, the user always reaches the initial screen, with [Exit \[Seite 492\]](#), "the screen calling the task". If data is not yet saved, a confirmation prompt must be displayed.



Navigation options to higher hierarchy levels (1)

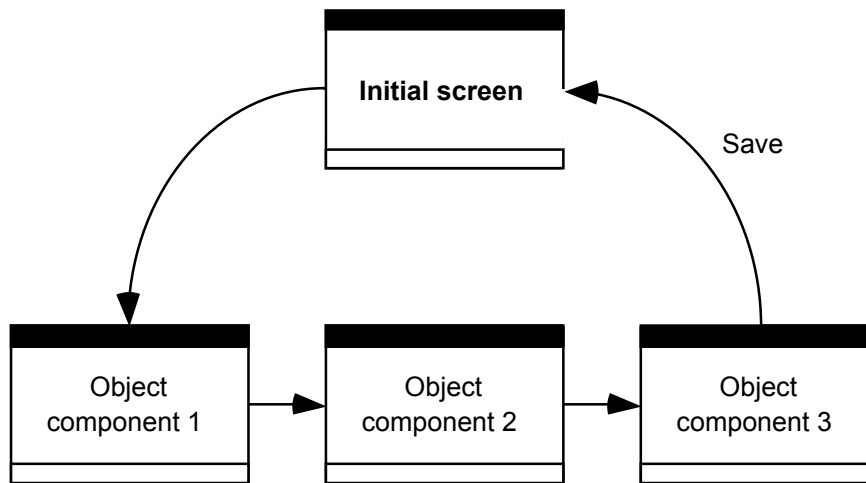


Navigation options to higher hierarchy levels (1)

- **Saving when making fast entries**

In some cases (for example fast entry), it can be useful to link the function [Save or Post](#) [\[Seite 524\]](#) with simultaneous navigation to the initial screen.

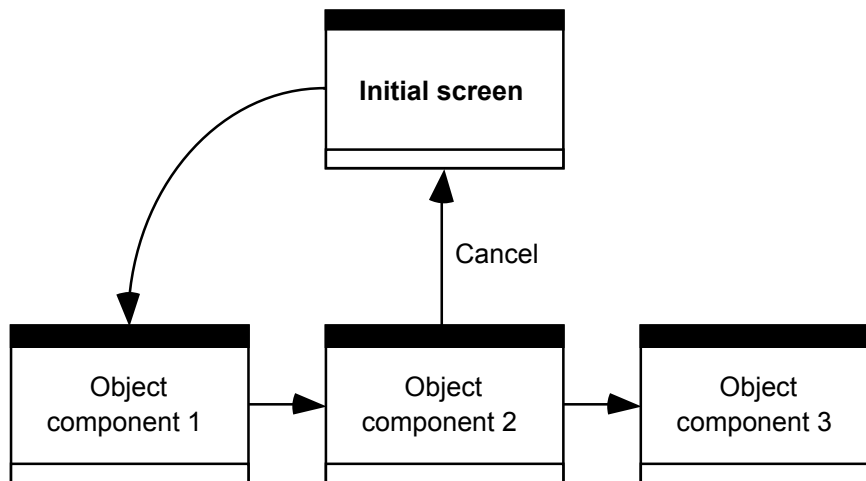
The Object Component Screen



Navigation to the initial screen when saving

- **Navigation in case of inconsistent data**

If navigation to a superordinate hierarchy level is impossible, for example, because data is still inconsistent, the [Cancel \[Seite 470\]](#) function is still active. In this case, however, data entered by the user will be lost.



Cancel is to be supported even when data is inconsistent



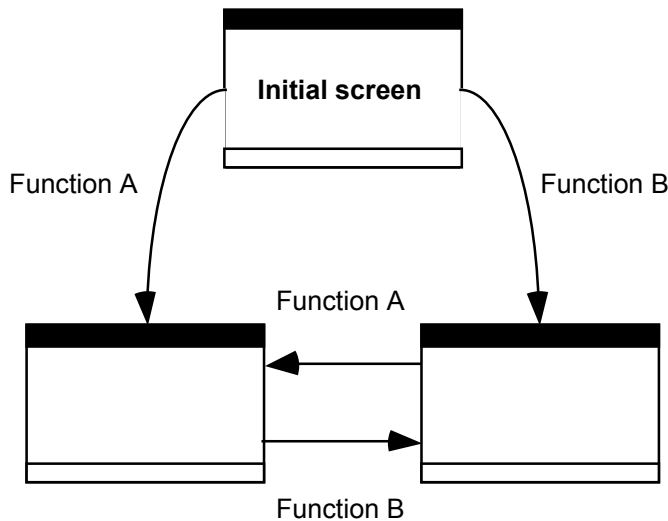
In R/2, the user has chosen *Create supplier*. On the second screen of the screen sequence, he wants to cancel the process. The data cannot be saved as it is not yet complete, but the user should be able to cancel the process in order, for instance, to return to the initial screen. (If data can be held, the function Hold should of course also be supported.)

Navigation Options on the Same Hierarchy Levels

- Choosing object components without restrictions

The Object Component Screen

If it was possible when accessing the object processing to choose the object component via various functions, then the same functions are also to be available for navigating on the same hierarchy level.



Same navigation functions at the same hierarchy level as for navigation at the lower hierarchy level

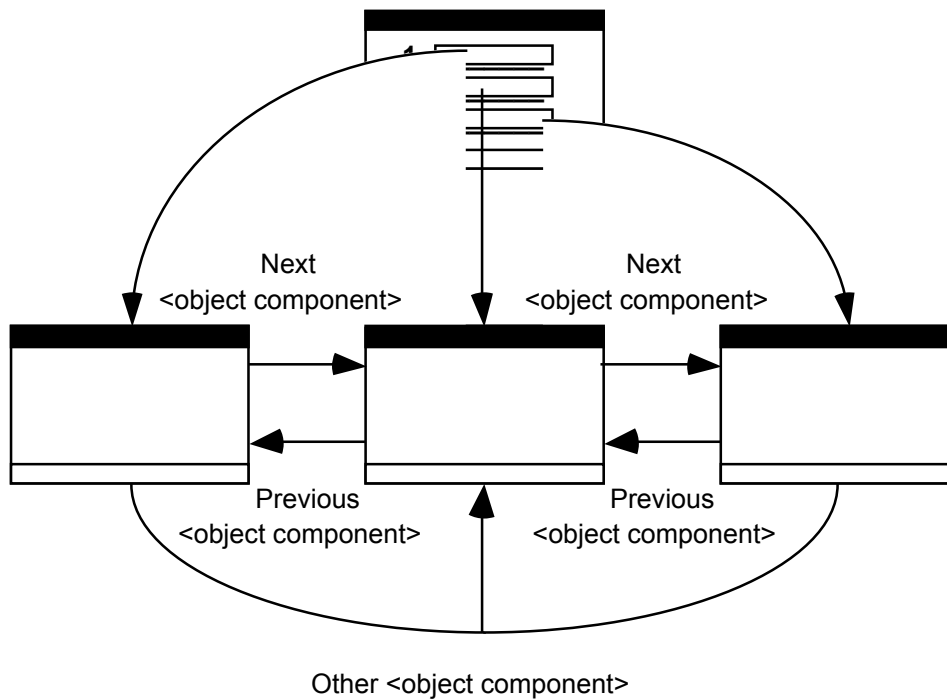


For the navigation between the same hierarchy levels in SAPoffice, you use the same functions that apply to the initial screen.

- **Access via object component list screen**

If the access to the object processing occurred via the object component list screen, then it is possible when processing an object component to branch with the function [Previous <object component> \[Seite 538\]](#) to the object component's predecessor (according to the list) and with the function [Next <object component> \[Seite 538\]](#) to the object component's successor (according to the list). With the function [Other <object component> \[Seite 476\]](#), it is possible to branch directly to any object component on the list. This does not have to be supported, however. With Other <object component>, it is not possible to branch to an object component outside of the list.

The Object Component Screen



Navigation when accessing via the list screen



In the case of the RP master data, the user has chosen the residence of a person on a list at a certain date. With Previous record, he receives the previous residence, with Next record, the next residence.

- Preselection**

If a selection has already been made by selecting something from the object component list, the functions only branch between these selected lines.



In a list of line items, the first, third and seventh item were selected, and then the first is displayed. With *Next item*, the selected item 3, the next item in the list, is displayed, rather than item 2.

- Nested lists/tables**

If the displayed screen in turn consists of a list/table via which the user can branch to a "component of object component" screen, then Previous <object component> and Next <object component> only navigate at the current level.



Please refer to Previous <object component>, Next <object component> (Functions) for an example.

- Changing the view for object components**

The Object Component Screen

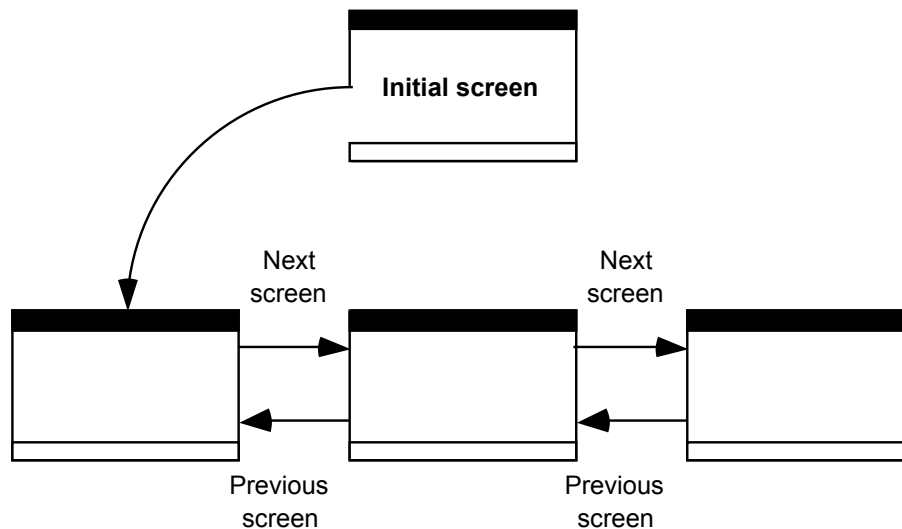
If you can change the view on subordinate screens - that is, if the user can look at yet further screens for an object component - then after selecting Previous <object component>, or Next <object component> he should see again the original view with which he initially accessed the display of the object components of the next displayed object component.



Please refer to Previous <object component>, Next <object component> (Functions) for an example.

- **Predefined sequence**

If the object components are concatenated in a predefined sequence, it is possible to branch to the previous screen with the function [Previous screen \[Seite 536\]](#) and to the next screen with the function [Next screen \[Seite 536\]](#).



Navigation in a predefined sequence

- It can also be possible to navigate to the next screen with [Continue \(Function\) \[Seite 540\]](#), if useful.



In R/2, a predefined sequence of screens is processed when creating a vendor. In this case, you should branch to the next screen with the function [Next screen \[Seite 536\]](#), for instance, from the accounting data to the purchasing data. With [Previous screen \[Seite 536\]](#), you would then branch from the purchasing data to the accounting data.

Navigation Options at Lower Hierarchy Levels

If the subordinate hierarchy levels can be chosen freely, any functions can be defined (see initial screen); if a choice was made from a list, then a distinction is to be made as on the object component list screen (see that topic).

The Object Component Screen

The Sub-Component of Object Screen

[Zurück \[Seite 481\]](#)

[Anderes Objekt \[Seite 474\]](#)

Function

Sometimes there are further subdivisions of an object component from which you can choose.



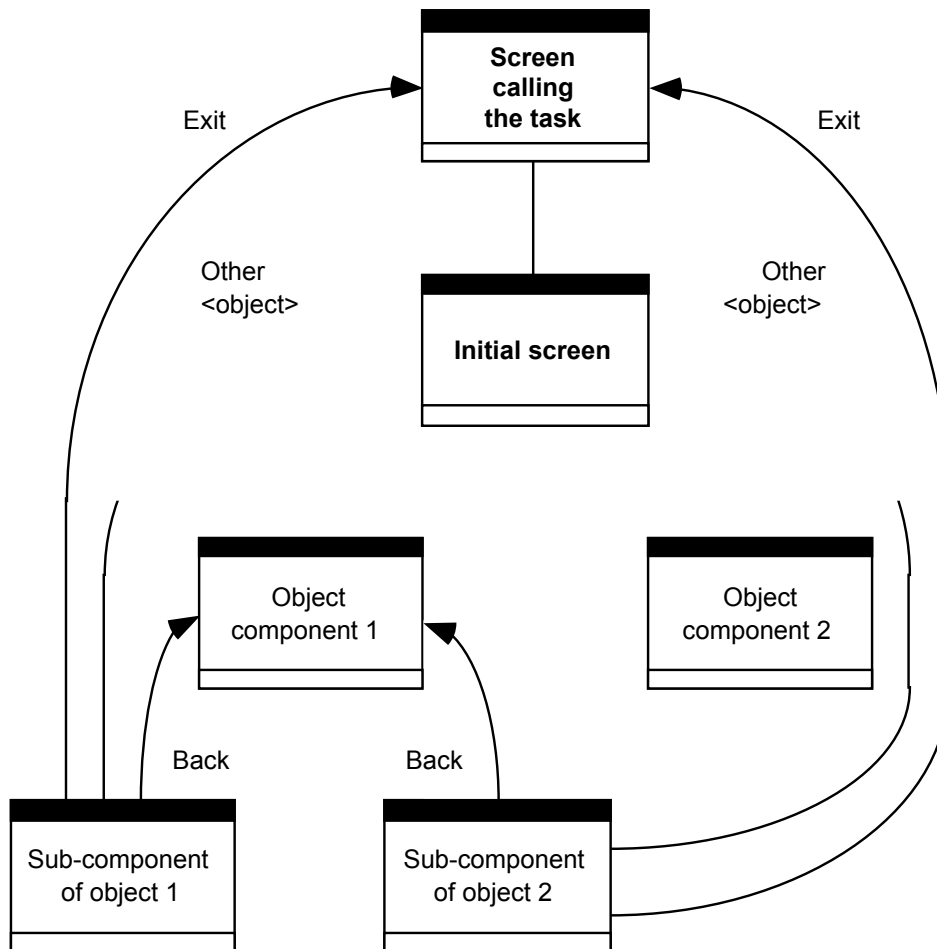
In RV, a sales and distribution document consists, for example, of the object component "Business data". The "Business data" is divided here into "Order", "Dispatch" and "Billing document".

Navigation Options at Higher Hierarchy Levels

- **Standard navigation**

With Back, the user returns to the object component screen, with Other <object>, to the initial screen, and with [Exit \[Seite 492\]](#) to "the screen calling the task".

The Sub-Component of Object Screen



Return options in the R/3 System

- **Navigation when data is inconsistent**

If it is not possible to navigate to a higher hierarchy level - for instance, because the data is not consistent at this point in time - the function [Cancel \[Seite 470\]](#) should still be supported. The data entered by the user, however, is lost in this case.

Navigation Options at Same Hierarchy Levels

The same rules as for navigating between object component screens apply.

Navigation Options at Lower Hierarchy Levels

If further hierarchy levels exist, the same rules as for navigating from an object component screen to a sub-component of an object screen apply, that is: If the subordinate hierarchy levels can be chosen freely, any functions can be defined (see initial screen); if the choice was made via a list, then the same distinction is made as on the object component list screen (see corresponding section).

Cross-References

Cross-References

- [Back \(Function\) \[Seite 544\]](#)
- [Cancel \(Function\) \[Seite 470\]](#)
- [Choose \(Function\) \[Seite 487\]](#)
- [Continue \(Function\) \[Seite 540\]](#)
- [Exit or End <application> \(Function\) \[Seite 492\]](#)
- [First Page, Last Page \(Function\) \[Seite 505\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Other <object> \(Function\) \[Seite 474\]](#)
- [Other <object component> \(Function\) \[Seite 476\]](#)
- [Previous Screen, Next Screen \(Functions\) \[Seite 536\]](#)
- [Previous <object component>, Next <object component> \(Functions\) \[Seite 538\]](#)
- [Save or Post \(Function\) \[Seite 524\]](#)

Object-Oriented User Interface

A feature which is common to all graphical user interfaces is that they all follow an object-oriented approach. This principle has only remotely to do with "object-oriented programming". An object-oriented user interface allows the user to handle objects as directly as possible without having to consider the particularities of the DP installation.

This object-oriented approach is reflected in the menu structure and in the object and action selection when the user processes objects in the work area.

[General Explanations \[Seite 130\]](#)

[Selecting Objects and Actions \[Seite 132\]](#)

[Cross-References \[Seite 133\]](#)

General Explanations

[Menüleiste der Anwendungsebene \[Seite 559\]](#)

The following example explains the object-oriented concept and its advantages:



Consider the user work area as a desk. There are various things on this desk such as pencils, file cards, papers and files. The employee can now take a pile of newspaper articles (object class) and go through the articles (objects), read them, make notes, staple them together, and sort them. The user might also add certain new articles to the pile. Then the user takes the card index (object class), takes individual cards from it (objects) and reads them or makes notes on them. Of course, new cards will be added from time to time.

Advantages of an Object-Oriented Approach

The employee is spared the continual opening and closing of the card index and the stack of papers. From the quantity of similar things compiled, the user only needs to work with some of them at a time. After the user has edited the file card and put the card index away again, he or she takes the pencils in order to sharpen some of them. Industrial productivity improvements are to a large extent the exploitation of this principle, whereby setup times, transport times and training periods, among other things, are reduced.

A second advantage is that the employee does not have to commit himself to particular actions when choosing the object class, but can carry out the appropriate action, depending on the characteristics of the object.

Disadvantages of an Object-Oriented Approach

The advantage of working with the current object class, however, becomes a disadvantage when the employee wishes to read an article and then to create a new file card for it straight away. The **work process**, so to speak, cuts across the object classes. In this case, the changes of the object classes required by the work process should be supported by specific navigation options. This adaptation of the interactive possibilities to the work process of the user leads to the concept of a process chain.

Conclusions for the Design

The decisive factor in gathering many objects in an object class is, thus, that it is generally possible to process a number of elements of this class one after another, using the same actions. The more effectively the objects with which an employee works are grouped into classes, the less he or she needs to "fetch" an object class for editing or to name it, and the less he or she therefore needs to rename the action. Ideally, the user needs only choose the function [Other object \[Seite 474\]](#) after editing a particular object, and enter the name of the new object.

The second advantage of an object-oriented procedure relates, however - in contrast to the procedure already outlined - to the processing of one and the same object by means of various actions, without the employee having to plan these long in advance of choosing the object. This should also be possible for the user, so that we now have to combine two potentially conflicting requirements:

- the possibility of choosing an action applicable to as many objects of a class as possible (= mass processing), and

- the possibility of choosing flexibly all the actions applicable to an object (= comprehensive, sequential object processing).

The object-oriented dialog has, thus, to be combined with a job-oriented and user-oriented design.

Relating Objects to User and Task

Inadequate analysis of the user's knowledge and of his work processes, an object-oriented interface design may force the user to work with objects in a form that he does not recognize. He frequently thinks on the basis of objects which originate from his work process and his workplace, not on the basis of objects which correspond to the abstract commercial or technical point of view.

For this reason, the objects should always refer to the working environment and the objectives of the user. They must be familiar and depict the conceptual framework of the user's work (and not the database system perspective, for instance). Processes such as "transfers" or "incoming payments" are thus also possible on the object class level. Concreteness or "physical reality" is not a defining attribute for an object.

The R/3 concept involves a move away from rigid process-based interaction by means of transaction codes to flexible, object-oriented user menu management.

These requirements are realized and standardized as far as possible in the pull-down menus of the SAP tasks under [Menus - General Structure \[Seite 548\]](#), Menu Bar at Task Level and [Menu Bar at Application Level \[Seite 555\]](#).

Selecting Objects and Actions

Object/Action Approach

The graphical user interface object/action technique is object-oriented. The user chooses an object (by mouse-click) and then directly manipulates it (by moving the object to another place, for example). This procedure is intended to be analagous to normal life, in which one first reaches out for an object and only then is in a position to work with it. Contemporary graphical user interfaces, however, only partially allow this direct manipulation of objects. A large part of the object processing occurs indirectly: The user chooses an object, it is displayed in a selected manner by the computer, and then the user chooses the action (from menus or pushbuttons) which he wants to perform on this object. The visual selection is a necessary confirmation for the user, so that he knows on which object an action is being executed.

Choosing the Object and Selection Method

The user always first **selects** an object. Objects can be selected in various ways. As a general rule, the user must be able to both select the object and choose the action either with the mouse alone or with the keyboard alone. The common form is:

- Single mouse-click on the object or
- position the **cursor** on the object or
- position the **cursor** on the object and choose the [Select \[Seite 516\]](#) function

Choosing an Action and Selection Techniques

After selecting the object, the user chooses the required **action** for this object. The actions in the SAP System are generally offered in the pull-down menus and can be chosen either with the mouse, with the arrow keys, or with access characters. Frequently-used actions are entered into the application toolbar and can also be chosen by pressing the corresponding function key. If the focus (that is to say, the cursor) is on a pushbutton (in which case, the key is emphasized by a box), the pushbutton or the action underlying, it can be initiated by pressing **ENTER**.

Default Action

For the most frequent function on an object, screen or dialog box (termed "default action"), there are ways of calling it faster than that just described. The user should be able to execute this accelerated call both by means of the mouse (double-click with the left key on object) and by means of the keyboard (**ENTER** or **F2** on the object). All of these various selection techniques must be supported.

Cross-References

- Einstieg in eine Anwendung
- [Menüleiste der Anwendungsebene \[Seite 559\]](#)
- [Accessing Tasks \[Seite 77\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- Menu Bar at Task Level
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Selection Methods - Overview \[Seite 95\]](#)

Referencing Objects

Referencing Objects

To create a new data record in the R/3 System, users often copy an existing data record or master record and modify it to suit their needs. Several reference modes are possible:

- **Duplicate**

Objects or object components from the same object class are used as a model to create a new object.

- **Entire objects**

Object 1 of the object class X is duplicated to object 2 of the same object class and then modified. Especially relevant for master data:



Material master record 1 --> material master record 2

- **Items of other objects**

Only selected parts (items) are copied from object 1 into object 2. This can be accomplished by immediately specifying the required items or by displaying a selection list. This case is especially relevant for **documents**.

- **Referencing**

Data from an object of one object class is copied to be used in an object of another object class.



Goods receipt for the purchase order or for the production order

[Duplicating Procedure \[Seite 135\]](#)

[Referencing Procedure \[Seite 139\]](#)

[Cross-References \[Seite 141\]](#)

Duplicating Procedure

[<Objekt> \(Menü\) \[Seite 574\]](#)

When duplicating, the term to be used is "Copy from".

Complete Copy of Objects

From the pull-down menu on the application level, the action [Create \[Seite 479\]](#) is chosen.

The functionality should not be split into *Create* and *Create by copying* because the user would be forced to make a decision at a too early stage in the process, and be led into modal creation dialog. It is more flexible for the user to choose between *Create* and *Create by copying* on the initial screen where the user can **always** specify the reference number of the model object.

On the full-page initial screen, the new object number is entered at the top (in the case of manual number assignment), along with other parameters. If there is enough room, you can use a block heading with the entry "Current" to improve the organization of the screen.

The object to be copied is specified in the lower part of the screen. This information is always separated from the current object information by using a group box with the heading *Copy from*:

The screenshot shows a SAP interface for creating a new object. It is divided into two main sections: 'Object to be created' and 'Object to be copied'.

Object to be created:

- Customer: 90767
- Company code: AFR1 (African Lion Inc.)
- Account group: KUNA

Object to be copied (Copy from):

- Customer: POLT (OLeary Enterprise)
- Company code: 0012

The object to be copied is specified in the lower part of the screen.

The user can proceed to the next screen by pressing **ENTER**. On the follow-up screens, further information on the object to be copied is displayed only if the corresponding fields were completed on the initial screen. The details of the duplicated object are copied without any further checks. The user must check and modify them manually in the new object. Highlighting the copied information (see [Highlighting \[Seite 603\]](#)) is recommended, provided no other data is highlighted and this is not all the information on the screen.

Duplicating Procedure

Copying Object Components

The action *Create* is chosen from the **pull-down menu** at application level, which implicitly also contains the option *Create by copying*.

The object to be copied and options for more specific selections (for example, item, plant, vendor, and so on) are offered on the **initial screen**. The selected data of the object to be copied (items, for example) is copied to the data screens of the current object with the function *Copy*.

Dialog Options

There are three possibilities, depending on the user's input:

- No information on the object to be copied entered:
 - **ENTER** = navigation to the first empty data screen.
 - Selection of the function *Copy* = navigation to the first empty data screen.
- The object to be copied was specified but no selection parameters (e.g. items):
 - **ENTER** = list of all positions for more specific selection.
 - Initiating the function *Copy* = copying all model data and navigation to the first filled data screen.
- The object to be copied and selection parameters specified:
 - **ENTER** = list of the selected items where the number of items can be further restricted.
 - Initiating the function *Copy* = copying the chosen items and navigating to the first data screen.

In the optional **selection list**, all items, or only those previously mentioned are selected:

Purchase requisition Doc. type

Purchase requisition items

Item	Material	Qty. requested	Un	Short text	Deliv. date	Mat. grp.
<input type="checkbox"/> 10	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/> 20	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/> 30	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/> 40	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Selection list for selecting items

In the pull-down menu [Edit \[Seite 565\]](#), the functions [Select/Deselect \[Seite 516\]](#), [Select all \[Seite 473\]](#) and [Deselect all](#) should be available for this purpose.

In the **newly created document**, the chosen items are entered and displayed consecutively.

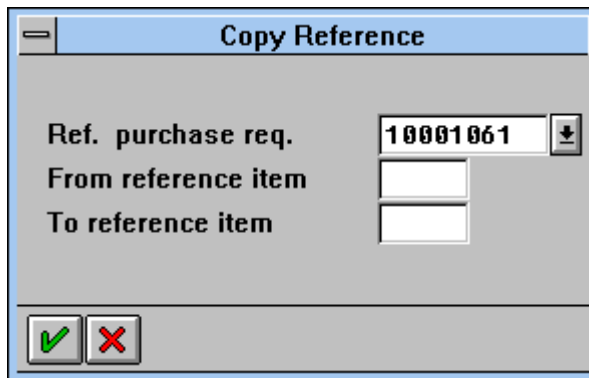
An additional object can be copied, if required. In this case, a dialog box is displayed on the initial screen with the same contents as the model block. The action for requesting this dialog box, *Copy from...*, is in the left pull-down menu <Object>.

Requisition <u>O</u> ther requisition <u>C</u> reate <u>C</u> hange <u>D</u> isplay <u>I</u> ndividual release C opy <u>f</u> rom... <u>S</u> ave <u>E</u> xit
--

Duplicating Procedure

Copy from... appears in the <object> = *Requisition* menu.

For some document types, the creation is generally made without copying an existing document. *Create by copying* may only be used in approximately ten percent of all cases. In these cases, the fields in which the user is to specify the object to be copied should **not** be on the initial screen. If required, it must still be possible to enter the relevant information with the action *Copy from...* in the pull-down menu or with the pushbutton *Copy from* in the following dialog box:

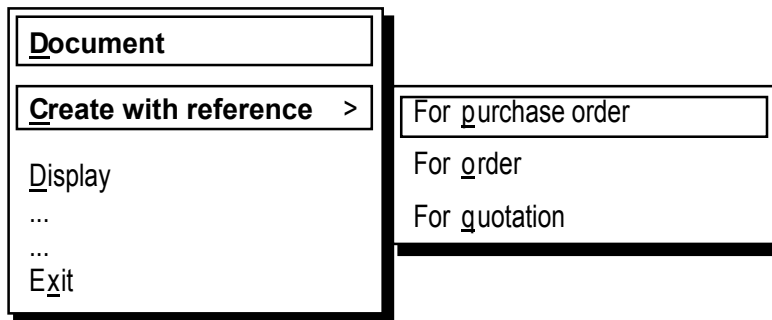
A screenshot of a SAP dialog box titled "Copy Reference". The dialog has a light blue header bar with the title. Below the header, there are three labels on the left: "Ref. purchase req.", "From reference item", and "To reference item". To the right of "Ref. purchase req." is a text input field containing the value "10001061" and a small square button with a downward arrow. To the right of "From reference item" and "To reference item" are empty text input fields. At the bottom left of the dialog, there are two buttons: one with a green checkmark and one with a red X.

Dialog box for entering copy information

Referencing Procedure

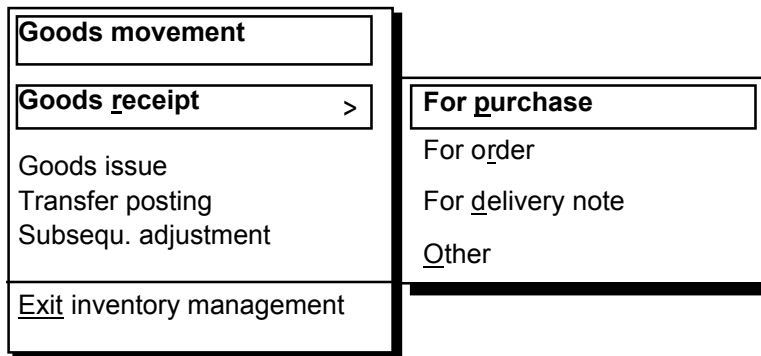
Please always use the word **Reference**.

At the second level of the pull-down menu at application level, the reference object is chosen:



Choosing the reference object

You can also specify the reference object's object class with pull-down menus without explicitly using the word *Reference*:



Specification of the object class of the reference object without using the term Reference

The number of the reference object (and other selection criteria, if required) is entered on the subsequent initial screen. The reference object is now the new object of another object class (in the example, material document).

The new document is usually a by-product of the main work process and a number is usually not assigned manually (internal number assignment is more likely). In contrast to duplicating, the input field for the new object should appear **after** the old object so that the user does not have to skip over it every time.

Cross-References

- [Create \(Function\) \[Seite 479\]](#)
- [Highlighting \[Seite 603\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Select/Deselect \(Function\) \[Seite 516\]](#)
- [Select All, Deselect All \(Functions\) \[Seite 473\]](#)

Searching in Structured Data

[Verweise \[Seite 151\]](#)

Generic Entries

In the R/3 System, the user can very often specify generic entries because the exact name of an object is not known or because the evaluation involves a large volume of objects.

When making generic entries, the user gives a vague description of the desired objects. For this type of search, the user can use **wildcard characters**.

Generic entries produce a large amount of related information from which the user can choose one or more objects.

Generic Search

Generic search denotes a search procedure which is triggered by a generic entry und which produces a hit list.

[Wildcard Characters \[Seite 143\]](#)

[Searching for Possible Entries \(without Matchcode\) \[Seite 145\]](#)

[Searching for Possible Entries Using Matchcodes \[Seite 147\]](#)

[Processing a Set of Objects \[Seite 149\]](#)

[Further Uses of "*" \[Seite 150\]](#)

Cross-References

Wildcard Characters

The Characters

The following generic symbols can be used:

Symbol	Description
*	0...n any character (= any character string, possibly blanks)
+	any character
#	escape symbol, which must be used if you want the following character to be interpreted 'literally'



The following table shows examples of wildcard character usage. If the entries in fields are interpreted differently in some cases (or due to technical restrictions), these are specified.

Entry	Case*	produces	Examples
A*	always	A...	A1, AB, ABCDE, A1B2C3
AB*	always	AB...	AB0, ABC, ABCDEF, AB12345
A*B	always	A...B	A1B, XYZB, AB
*B	always	...B	B, AB, 123456789B
A+C*	always	A.C...	ABCD, AACCCDD, A1C3
A#C*	always	A*C...	A*CA, A*CB
*	always	all	
<string>	case 1	all	
or	<string>	<string>	
Blank	case 1+2	all	
	or	blank	blank
#	always	blank	blank

*) Cases 1 and 2 see below.

Case 1: Searching for possible entries (no matchcode)

The generic search behaviour of input help differs in the cases “<string> or nothing (blank)” from that in other application areas. This special procedure was chosen because the existing entries in a field usually have no relevance for input help. Often the user triggers the F4 search to correct a previously made entry.

Case 2: Report selection screens

The “*” character cannot be used with type P fields. To be consistent with other fields, the blank character for “Select all” is used on report selection screens.

Wildcard Characters

Using Special Characters in Object Names

The applications should try to restrict the use of "*" and other special characters in names (for example, the name "A*" should not be allowed for a document).

Escape symbol

If you have to use "*" in names, you can address objects containing this character with the escape symbol "#":

Entry: A#*30 for object name: A*30

Blanks

If an input field explicitly requires entry of a blank character, the user cannot enter this, because it cannot be distinguished from the blank character used for the "Select all" search. In this case, the escape symbol is to be used.

Reason: This entry applies to one-character input fields, but cannot be distinguished from the correct character string "#_" with multiple-character fields. ("_"= blank character).



In the case of generic entries with an escape symbol, the input field may not be long enough. If such entries are expected, a scrollable input field should be defined as of Release 3.0. The visible length of the scrollable field corresponds to the length of the non-generic entry.

Searching for Possible Entries (without Matchcode)

Possible entries functionality (without matchcode)

You can only realize generic search to determine field input values by using the [Possible Entries \(Function\) \[Seite 499\]](#) functionality (that is, the F4 function key and/or the [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#) next to the field and/or the button in the [Standard Toolbar \[Seite 307\]](#)). The ENTER key, other function keys or menu options should not be used. You can no longer display all possible entries values by entering "*" in an input field.

Procedure

1. The user does not know a term in a given task, makes a generic entry to find a set of possible entries and chooses F4 (or the corresponding functionality).
2. After choosing a value from the list, the user returns to the previous screen.
3. The user completes other fields and activates the desired action.

That is: the possible entries functionality makes an "excursion" and then returns to the previous screen.

Interrupting the Search

The search process is interrupted after a certain number of hits (determined by the application). The system displays a dialog box which allows the user to further restrict the search. This procedure is repeated until less than the specified number is found or the user does not wish to further restrict the search.

The search should be interrupted after the system has found 100 items. With time-intensive search processes, this value can be smaller, but an interruption value of greater than 100 is not allowed.

The search is to be interrupted in all F4 help functionalities including self-programmed F4 help features.

Initial Screens

On initial screens, the user specifies objects for further processing. The object name can be entered exactly or generically. If the user does not know the exact name, the F4 functionality should be available to find the correct object name. This contributes to a consistent structure and use of initial screens in the R/3 System. In addition, initial screens behave in the same way as other screens.

Procedure

On initial screens, the procedure is as follows:

1. The user specifies an object.
 - 2a With ENTER: The system goes to the specified object or does not react because the object could not be identified. In the second case, the system displays a message in the status bar.
 - 2b With F4: A generic search process is triggered. The system displays the result in a hit list from which the user can choose an object.

Continue with 2a -> F4 makes an "excursion".

Searching for Possible Entries (without Matchcode)

If required, the user should be allowed to branch to a selection screen for a more precise pre-selection of objects. (This is to be decided by the individual applications).

Searching in Non-Key Fields

When searching in non-key fields, there may be identical entries and the system may find several items although the user has made fully specified input. In this case, you should proceed as with generic entries, although the user has not entered a wildcard character.

Searching for Possible Entries Using Matchcodes

With some fields, the user can use matchcodes to search for possible entries. Matchcodes are pre-defined combinations of secondary keys which facilitate the search for possible entries.



To issue an invoice, the user requires the customer number. To find the unknown number, for example, the user can search by name and surname of the customer (one matchcode) or by other non-unique combinations of customer data (other matchcodes) which are already known. If the system finds several data records, it displays a list from which the user can choose the desired customer number.

Procedure

1. The user does not know a term on a task screen. However, a matchcode is available and the user chooses **F4** (and/or the corresponding functionality).
2. The system displays a list with matchcodes from which the user can choose.
3. The system displays a dialog box with secondary keys for which matchcodes are available. The user completes the secondary key fields; generic entries are possible. The user then chooses **ENTER**.
4. The system displays another list from which the user can choose the desired entry value. The user then returns to the original screen.
5. The user completes other fields and activates the desired action.

Entering Matchcodes Directly

If the user knows the matchcode, this can be entered directly in a field:

=<matchcode ID>[.<field 1>[.<field 2> ...]...]

Wildcard characters can be used.

Procedure

1. The user does not know a term on a task screen for which matchcodes are available.
2. The user enters a (known) matchcode in the matchcode field. Wildcard characters can be used.
3. The system displays a list from which the user can choose the desired entry value. The user then returns to the original screen.
4. The user completes other fields and activates the desired action.



Currently, there is an implicit * at the end of the matchcode entry, that is, the entry **"A"** produces the same result as **"A*"**.

Default Matchcode

If a field has a default matchcode and the user chooses **F4**, the system goes directly to the selection list of the corresponding matchcode.

Searching for Possible Entries Using Matchcodes**Indicating Input Field For Which Matchcodes Are Supported**

Input fields which allow the use of matchcodes have a platform-specific identification. The user can easily distinguish between fields for which matchcodes are available and fields for which lists with possible entries are supported.

Processing a Set of Objects

A generic entry cannot only be used to specify a list of data but also a set of objects to be processed. In this case, the system displays a screen which contains exclusively fields used to identify the objects to be processed. The user triggers processing by making a generic entry on this screen and choosing ENTER.

Procedure

1. The user makes a generic entry to choose certain objects for processing.
2. The set of chosen objects can be further restricted, if required.
3. The objects are then processed in parallel or in sequential order.

Further Uses of “*”

Further Uses of “*”

[Maskierungszeichen \[Seite 143\]](#)

The “*” character can also be used for other purposes, if it does not conflict with a wildcard entry.



- Display of automatic calculations

The use of “*” in object names should be restricted (see section on wildcard characters). The “*” as indication of already processed objects should be replaced with a status display as of Release 3.0.

Cross-References

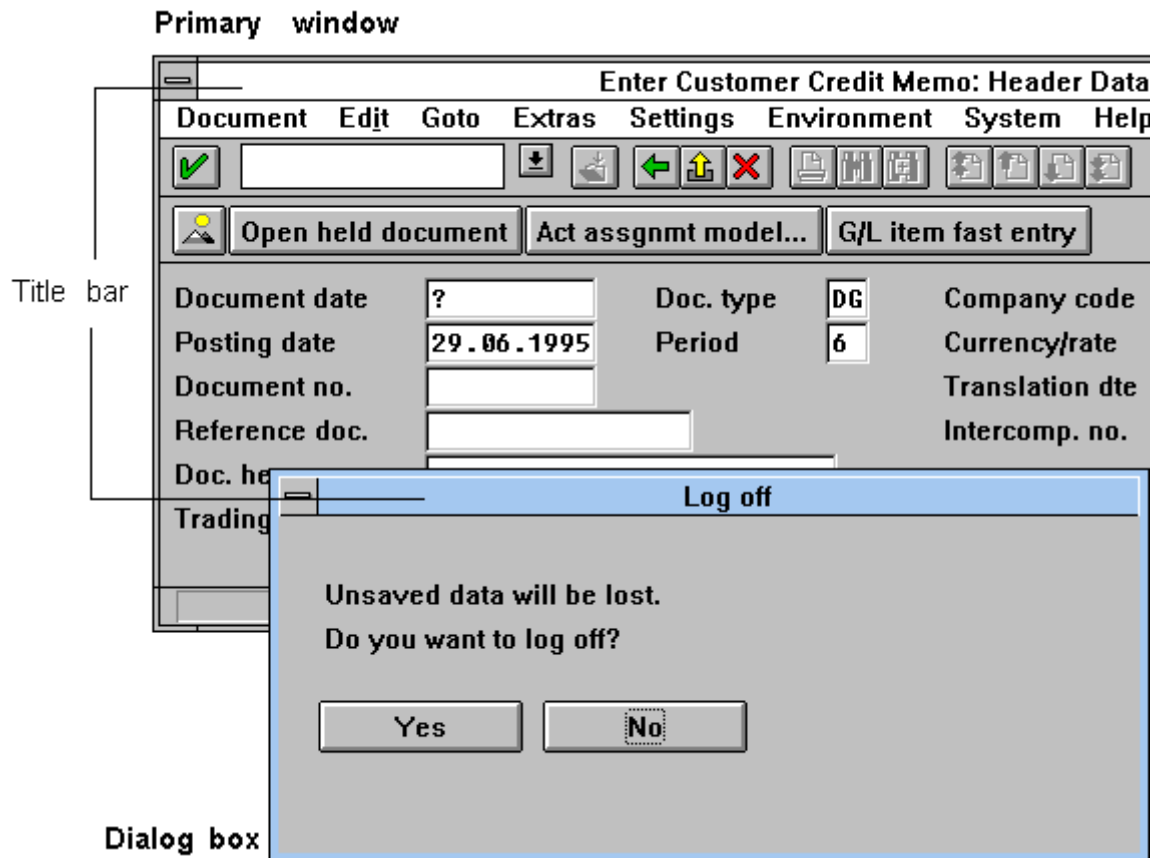
- [Possible Entries \(Function\) \[Seite 499\]](#)
- [Standard Toolbar \[Seite 307\]](#)
- [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#)

Title Bar

Title Bar

The title bar is the topmost line of primary windows and dialog boxes. The title of a window is a unique label which identifies the location of the current window in the system and thus facilitates efficient navigation. Besides identifying the current task, the title also identifies the contents of the window.

Illustration



Title bar in the primary window and the dialog box

[General Guidelines \[Seite 154\]](#)

[Guidelines for the Various Levels \[Seite 155\]](#)

[Naming General SAP Screens \[Seite 157\]](#)

[Title Bar in Dialog Boxes \[Seite 161\]](#)

[General Notes \[Seite 164\]](#)

[Technical Notes \[Seite 165\]](#)

[Cross-References \[Seite 166\]](#)

General Guidelines

General Guidelines

Where Am I?

The title bar must locate the current screen in the system. The action which the user executed to arrive at this point is displayed on the current screen.

Where Did I Come From?

In some situations, context is to be provided to establish the connection to previous steps. The title should answer the question: "Where did I come from?" or "In which higher-level action am I currently located?"

Guidelines for the Various Levels

To access a particular task in the R/3 System, the user has to navigate through the system hierarchy. The titles of the various screens can help to make this hierarchy transparent. The title should contain the name of the various system levels.

The following section specifies the screen titles (if only one screen exists on a particular level) and/or provide design guidelines and examples (if several screens exist on a level) which you can use to identify the various levels of the system hierarchy (see [Levels in the SAP System \[Seite 72\]](#)).

Conventions

As far as the **syntax** of the following design guidelines is concerned: Variable terms which must be replaced by a full identification appear in angle brackets. Expressions in square brackets are optional. After "Title:", examples are in quotes.

Main Menu Level

Title: *SAP R/3*

Application Level

- If the user gets to this screen by choosing options from several cascading pull-down menus, the title of the screen displays the last chosen menu option (current menu option).

Syntax: *<current menu option>*

Title: *Accounts Payable*

Inventory management

You can also create the title of the screen from the concatenation of the higher-level menu option and the last chosen menu option. This, however, is only allowed where it helps to avoid ambiguities.

Syntax: *<higher-level menu option> - <current menu option>*

Title: *Customizing - Extended General Ledger G/L*

Task Level

The title bar used to identify a task consists of three components:

- the name of the higher-level task or object class (ObjCI),
- the chosen action, which refers to the chosen task or the object class as a whole (act),
- optionally, the description of the current screen (screen) and - in some cases - an object (if identifiable), separated by three blanks, consisting of the field name and value (obj.)



In an editor, this would be the file name; for a posting document, for example, "document number 25".

If the current screen is reached from the previous system status by choosing a menu option or activating a pushbutton, the *<screen>* can also be created directly from the

Guidelines for the Various Levels

name of the menu option or the pushbutton text. You can also specify the path which brought the user to the current screen (in the form <screen-2>-<screen-1>-<screen>).

Syntax: <Act> <ObjCl>[: <Screen>[<Obj>]]

or <Act> <ObjCl>: [<screen-2>-][<screen-1->]<screen>

Title:

- Create unit costing: Access
- Create unit costing: Item list
- Display FiAc document: Initial posting document

number 1674

When naming **reports**, you must be aware that the user generally calls the report by choosing a corresponding menu option. The name of the menu option should be the same as the title of the report, if possible. They should at least contain the same word components.

Syntax: <ObjCl>: <screen>

Title: *Document compact journal: Result*

Profitability Analysis: Selection



The **title bar of a report** must be distinguished from any **list headers** which are located within the work area above the report listing. In the R/3 System, the title bar is always located **above** the menu bar (an orientation for the user).

Naming General SAP Screens

[Erfassen \[Seite 503\]](#)

[Einzelerfassung \[Seite 503\]](#)

[Bearbeiten \[Seite 529\]](#)

[Einzelbearbeitung \[Seite 529\]](#)

[Sammelbearbeitung \[Seite 529\]](#)

[Zusätze \[Seite 567\]](#)

This section describes and assigns names to screens that appear repeatedly in the SAP System **task level**. A title bar and examples of the entire title are specified for each of these screen types as far as possible. As described earlier, the title bar at this level generally consists of the following components:

- `<action> <object class> [: <screen>[<object>]]` or
- `<action> <object class>: [<screen-2>][<screen-1>]<screen>`

General Guideline

Avoid redundancy and text components that do not provide further information. Thus, try to do without "screen" and "data" if the title is otherwise clear enough.

Specific names have been assigned to **general SAP screens** in order to ensure consistent names for the component `<screen>` at **task level**:

Initial Screen

The first screen of a task from which you request the object you want to edit or to create. You generally reach this screen

- by choosing a menu option
- by entering a path name (fast path)
- by entering a transaction code

This also applies to the initial screens of programs that behave like a task.

`<screen> = "initial screen"`

Title:

- Enter document: Initial screen
- Create material: Initial screen

Overview Screen

A screen within a task which displays object components (mostly items) of an object grouped together in table form. For each object component, there is usually also a detail screen.

`<screen> = "Overview"`

Title:

Naming General SAP Screens

Goods receipt for the purchase order: Overview

Special case:

Overviews can display several levels of detail. These levels should be specified by an additional identifier. Do not replace the term "overview" with another word.

Title:

- ...: Item overview
- ...: Overview open items
- ...: Overview selected items

The definitions of **Detail screen**, **Data screen** and **Entry screen** differ slightly. You use one of them depending on which aspect you want to emphasize.

Detail Screen

A screen within a task where data for **one** object component is displayed in detailed form (as opposed to overview).

`<screen> = "Details"`

Title:

Display table: Details

Data Screen

A screen within a task where **data of a particular type** for an object or object component is **displayed or changed**.

Here, the emphasis is on the type or the contents of the data.

`<screen> = <specific contents of the screen>`

Title:

- Change purchase order: Header
- Change material: Purchasing

`<screen> = "General data"`, if the content is not specific

Title:

Change material: General data

Entry Screen

A screen within a task which is mainly used to **enter** data for an object. "Entry screen" is the generic term for "Single entry screen" and "Multiple entry screen".

On entry screens, the user should also be able to change data. Although the title bar says "Entry screen", this also includes the change functionality.

Here, the emphasis is on the screen design: The user should be able to enter data as efficiently as possible.



In this context, you should read the descriptions of the [Enter, Single Entry, Collective Entry \(Functions\) \[Seite 503\]](#) and Edit, Single processing, Collective Processing functions.

Single Entry Screen

Screen within a task where the user enters data of **one** object component (generally completely).

<screen> = "Single entry"

Title:

Create goods receipt: Single entry

Multiple Entry Screen

Screen within a task where the user enters data for **several** components of an object (generally not completely; only the essential data).

<screen> = "Multiple entry"

Title:

Create goods receipt: Collective entry

Entry vs. Edit

If the [Create \[Seite 479\]](#) and [Change \[Seite 477\]](#) functions occur in two separate tasks and these tasks have both a single entry and a multiple entry screen, you can formulate application-specific title bars.

Title:

- Create <object>: Single entry
- Create <object>: Collective entry
- Change <object>: Single change
- Change <object>: Multiple change

Generally, you distinguish between the single entry and multiple entry method. In this case, the entry method is specified in the task name and not in the screen name. (Note: In the following two examples, screens were used whose title bars are strictly speaking redundant. For reasons of consistency, however, the title bar for these screens should nevertheless also be listed.)

Title:

- <object> single entry: Details
- <object> collective entry: Overview
- <object> single change: Details
- <object> multiple change: Overview

If the entry and the change method are grouped together in a task (that is, there is only one processing method) and the emphasis is on the change option, the task name is correspondingly changed to *Edit*.

Naming General SAP Screens

Title:

- <object> single processing: Details
- <object> collective processing: Overview

Additional Data Screen

A screen within a task which contains additional data for an object or object component and which is generally called by choosing a menu option from the Extras menu on a data screen.

<screen> = <menu option calling the screen or dialog box>

Title:

- Dialog box examples (see below):
- Details: Short text
- or shorter: Administration data
- Primary window example:
- Create material: Administration data

Selection Screen

A screen of a report clearly identified as a prompt on which you specify the criteria to define the information you want in the report.

<screen> = "Selection"

Title:

ABC analysis: Selection

Result Screen

A screen of a report which displays the result of the report and which can possibly be used as the basis for further analyses.

<screen> = Result

Title:

ABC analysis: Result

Other screens which do not fit into above categories are possible, but should be named and used in line with the guidelines in this section.

Title Bar in Dialog Boxes

Dialog boxes can be displayed in various forms. The contents of the title bar varies correspondingly. The system places the header in the center of the first line of the dialog box. (For further information: see [Modal Dialog Box \[Seite 65\]](#).)

After Pressing F4 (Possible Entries)

The help system generates these titles automatically and enters them into the dialog box. If the developer programs the F4 functionality, he or she has to set the title conforming to the following rules.

It is displayed that the dialog box was called via F4, thus, that the literal "PossEntr" and the field name (FldID) are requested for the possible entries. If an additional dialog box is initiated from the first one (this case is also to be programmed by the developer), the value selected in the first window is copied to the title bar (Sel) and separated by three blanks.

Syntax:

PossEntr: <FldID>[<Sel>]

Title:

PossEntr: Process

For an additional dialog box following the first one:

PossEntr: Process balance sheet adjustments

The "PossEntr" information remains constant, the field name results from the name of the field for which the Possible entries function was called.

After Choosing a Menu Option from a Pull-Down Menu

The options chosen from the menu bar (optional) and the pull-down menu are listed. The entry chosen from the menu bar (menu) is placed at the beginning (where this aids comprehension). In addition to the first pull-down menu (1. PM), the second pull-down menu (2. PM) and the third pull-down menu (3. PM) are displayed, if available, to indicate the path taken.

Syntax:

[<menu>:]<1. PM>[-<2. PM>][-<3. PM>]

Title:

- Insert
- Copy line
- Goto: Graphic

If the title bar does not describe the window adequately and/or the title is too long, even though you have followed above design guidelines, you can define the title by specifying an optional action (act.) and the affected object (obj.):

Syntax:

- [<Act>] <Obj>

Title:

Title Bar in Dialog Boxes

- Specify date format

If a Second Dialog Box is Called From a Dialog Box

The structure is similar to the one described in the previous section; in addition, a freely-selectable one-word title (tit) for the second dialog box is displayed. Alternatively, a value selected from a list in the first dialog box can be displayed (sel).

Syntax:

- <1. PM>[-<2. PM>][-<3. PM>]: <Tit>
- <1. PM>[-<2. PM>][-<3. PM>]: <Sel>

Title:

- Copy line: Parameter
- Copy: Letter1

If a [<Act>]<Obj> title was defined in the first dialog box (see "After choosing a menu option from a pull-down menu"), the title can appear before the current one-word title (tit) in a follow-up dialog box:

Syntax:

[<Act>]<Obj>:] <Tit>

Title:

Assign bank transactions: Bank charges

Title Bar for Dialog Boxes Displaying Messages

In certain situations, the program can display dialog boxes which contain error messages, warnings, information as well as confirmation prompts for the user.

The title bar should refer to the step which the user last executed and which caused the message to be output.

The action (act.) last executed may optionally be specified:

Syntax: [<Act>] <EdObj>

Title: Exit transfers

Delete line

The titles for some functions which cause returns and initiate a dialog box are to be specified as follows:

Exit

Title: "Exit <entire object>"

Cancel

Title: *Cancel* <object component>

Back

Title Bar in Dialog Boxes

Title: *Back*

Other <object>

Title: *Other <object>*

Create, Change or Display

Title: Create <object>

Change <object>

Display <object>

General Notes

General Notes

- For the individual text components, follow the guidelines described in [Abbreviations \[Seite 593\]](#).
- Keep to the maximum possible length of a title bar (60 characters). Please note that the translation into another language may require more characters (see [Translation \[Seite 621\]](#)).
- Choose the freely-selectable terms appropriately: They should describe options from the user's point of view and not from technical considerations.
- Take blanks and the punctuation marks ":", "-", "(", and ")" into account.
- Note for translators: In some languages, the <Action> <ObjCl> order may have to be reversed.

Technical Notes

Static Title Bar

The developer has to set the titles in the ABAP/4 coding himself using the command "SET TITLEBAR name WITH a b c d". The actual title is defined in the Menu Painter following the syntax and semantic guidelines specified in this standard.

Title Bar Set Dynamically

Titles can also be set in dialog modules and function modules. Alternatively, the title can be left empty and passed to the dialog module or function module as a parameter with the "&" variable.

Title Bar for Reports

The title bar in report output can either be defined directly once using SE38 or alternatively in the Menu Painter. The title is then activated with SET TITLEBAR. The second alternative allows you to set the title bar in reports for different list screens in the same way as title bars in tasks.

Centered Automatically

The system centers the header both in the primary window and in dialog boxes.

Translation

Please make sure that the current length of the title allows a translation into other languages.

Cross-References

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Edit, Single Processing, Collective Processing \(Functions\) \[Seite 489\]](#)
- [Enter, Single Entry, Collective Entry \(Functions\) \[Seite 503\]](#)
- [Formulations \[Seite 601\]](#)
- [Hyphenation \[Seite 620\]](#)
- [Levels in the SAP System \[Seite 72\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Modal Dialog Box \[Seite 65\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Translation \[Seite 621\]](#)

Control Elements

[Tabellen und Listen \[Seite 339\]](#)

This section describes the SAP GUI control elements.

The introduction summarizes the control elements and their use. Then the individual control elements are discussed in detail, in alphabetical order.

[Control Elements - Overview \[Seite 168\]](#)

[Access Characters \[Seite 184\]](#)

[Application Toolbar \(Control Element\) \[Seite 227\]](#)

[Checkboxes \(Control Element\) \[Seite 176\]](#)

[Check Text \(Control Element\) \[Seite 300\]](#)

[Command Field \[Seite 201\]](#)

[Field Names \(Control Element\) \[Seite 324\]](#)

[Fields - Positioning \[Seite 253\]](#)

[Function Keys - Overview \[Seite 267\]](#)

[Function Key Menu \[Seite 278\]](#)

[Group Box \(Control Element\) \[Seite 280\]](#)

[Input/Output Fields \(Control Elements\) \[Seite 238\]](#)

[Long Screens \[Seite 290\]](#)

[Menus \(Control Elements\) \[Seite 295\]](#)

[Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#)

[Pushbuttons With No Fixed Position \(Control Element\) \[Seite 204\]](#)

[Radio Buttons \(Control Element\) \[Seite 193\]](#)

[Scroll Bars \(Control Element\) \[Seite 202\]](#)

[Standard Toolbar \[Seite 307\]](#)

[Status Bar \[Seite 305\]](#)

[Status Display \(Control Element\) \[Seite 302\]](#)

[Tab strips \[Seite 314\]](#)

Control Elements - Overview

Control elements are visible parts of the user interface which inform the user about the status of the system and allow him or her to initiate actions and to process R/3 System data.

Classification of the Control Elements

The following types of control elements are distinguished in the SAP GUI:

- Input/output elements
(input/output fields, field names, short descriptions, tables (input and read-only tables), lists, graphics, texts)
- Action elements
(function keys, pushbuttons with textual or graphical labels)
- Menus
(menu bar, pull-down menus, function key menu)
- Selection elements
(radio buttons, checkboxes, selection lists, etc.)
- Adjustment elements
(scroll bars)
- Organizational elements
(group box with group heading)
- Complex control elements (Widgets)
(table control)

[Types of Control Elements \[Seite 169\]](#)

[Using Control Elements \[Seite 173\]](#)

[Cross-References \[Seite 175\]](#)

Types of Control Elements

[Drucktasten \[Seite 711\]](#)

[Auswahlknöpfe \[Seite 193\]](#)

[Ankreuzfelder \[Seite 176\]](#)

[Auswahltabellen und -listen \[Seite 362\]](#)

[Auswahltabellen und -listen \[Seite 362\]](#)

[Blätterleisten \[Seite 380\]](#)

[TabStrip \(Register\) \[Seite 314\]](#)

Input/Output Elements

Input/output elements are used

- to display information, which the user can possibly change
- to receive from a small to a very large amount of information from the user

Output elements only display information. The user cannot change the data or only under particular circumstances.

Input/Output Fields

[Input/Output Fields \(Control Elements\) \[Seite 238\]](#) (I/O fields) are text fields in the work area of a screen. The user can enter data in I/O fields or the system can display information. I/O fields are defined in the Screen Painter.

Field Names

[Field Names \(Control Element\) \[Seite 324\]](#) are text fields (output fields) which only occur in association with input/output fields. They identify the function of the particular fields.

Short Descriptions

Short descriptions are text fields (output fields) which describe the contents of I/O fields in more detail.

Command Field

The [Command Field \[Seite 201\]](#) is an input field which allows the user to enter a typed command. This field is integrated into the standard toolbar (after the ENTER pushbutton) and implemented as a drop-down list which lists the last 20 commands entered by the user (command list).

You can enter fast paths or transaction codes in the command field.

Status Bar

The [Status Bar \[Seite 305\]](#) displays system messages and informs the user about the system status (for example, indicating the write mode, the system and the session). The status bar is located at the bottom of the primary window.

Types of Control Elements

Action Elements

Action elements initiate an action when the user chooses a function. The action can be performed on the current screen or cause a screen change.

Pushbuttons

Pushbuttons are fields with a textual or graphical label and initiate a particular action, if activated.

In the R/3 System, you find pushbuttons:

- in the standard toolbar (with symbol)
- in the application toolbar (with function name and the respective function key)
- in the work area of a screen at no fixed position (with textual or graphical label)

Pushbuttons can be activated with the mouse (single mouse-click) or the keyboard (exception: the pushbuttons in the standard toolbar cannot be activated with the cursor, only with the function keys assigned to them). Pushbuttons in the application toolbar and the standard toolbar are defined in the Menu Painter. Pushbuttons with no fixed position are defined in the Screen Painter.

Function Keys

[Function keys \[Seite 267\]](#) also initiate an action, but can only be activated with the keyboard. Currently, the R/3 System supports 24 function keys which are largely standardized. Some function keys are represented as pushbuttons in the application toolbar or in the standard toolbar.

All active function keys are displayed in the [Function key menu \[Seite 278\]](#). To display the menu, click the right mouse button.

Menus

A [menu \[Seite 295\]](#) is a list of items (for example, functions or attributes) which the user can choose one at a time. In this respect, they can be compared to pushbuttons and function keys.

Menu Bar, Pull-Down Menus

We distinguish between the menu bar and pull-down menus. The menu bar is a horizontal bar just below the title bar of the window. A pull-down menu is a submenu opened by choosing a menu item (known as cascading menus, if called from a pull-down menu). Pull-down menus are arranged vertically.

Besides using the mouse, you can activate menus with the [Menu Bar \(Function\) \[Seite 518\]](#) (F10) function. You select a menu option with the arrow keys and press ENTER to activate it.

Access characters

To access a menu directly, you can define access characters which are pressed in combination with the ALT key. Another direct method is the fast path. The user enters a period followed by a string of [Access Characters \[Seite 184\]](#) in the command field.

Selection Elements

Selection elements are control elements with which the user can make choices. Strictly speaking, action elements are also selection elements because they choose exactly one action. In this context, the term "selection element" refers only to the selection of objects, attributes or

Types of Control Elements

options. Different types of selection elements are used for different purposes (choosing one or more elements, the number of alternatives, etc.).

Radio Buttons

Radio Buttons (Control Element) choose exactly **one** item from among **several** alternatives. The number of alternatives is fixed and all items should fit on one screen (exception: vertical radio buttons in single-selection lists (step loops)). One option is always selected.

Checkboxes

Checkboxes (Control Element) allow the choice of **as many items as desired**, or none at all. The number of items is **fixed** and all items should fit on one screen (exception: vertical checkboxes in multiple-selection lists (step loops)).

Single-Selection Lists

Single Selection Lists contain at least two entries. The length of the list, however, is usually **not fixed** and all of the alternatives need not fit on one screen. The user cannot choose more than **one** entry.

Multiple-Selection Lists

Multiple Selection Lists contain at least two lines. The length of the list, however, is usually **not fixed** and all of the alternatives need not fit on one screen. The user can choose **several** entries.

Other Control Elements

Control elements can be used for various purposes. They can be used to choose a value from a value set, for example, to move a slider on a scale. This is possible for continuous as well as for discrete values. Typical control elements are the scroll bars which provide a means for viewing part of an area which is larger than the screen work area. Elements which show or hide information are more usually referred to as pushbuttons.

Scroll Bars

Scroll bars inform the user that there is more information available than is displayed on the current screen. They allow scrolling up or down in the information. There are vertical and horizontal scroll bars.

Further Controls

Further controls such as sliders, scales, and so on are currently not available in SAP GUI.

Organization Elements

Organization elements are used to provide visual grouping of related control elements on a screen.

Group Box

[Group boxes \[Seite 280\]](#) are used to group I/O fields with field names, or radio buttons or checkboxes. Group boxes consist of a rectangular frame and are identified by a group heading.

Types of Control Elements

Complex Control Elements

You can use simple control elements to develop more complex control elements. Complex controls offer more functionality, save developers to perform detailed work, and simplify interaction with the applications. Examples for complex controls are the table and the notebook. The notebook is currently not available yet in the R/3 System.

Table (Table Control)

The [Table Control \[Seite 340\]](#) feature allows to display tabular data structures in the R/3 System as known from table calculation programs. It supports functions, application developers had to implement themselves. It also allows to create entry, display, and selection tables.

Using Control Elements

[Drucktastenleiste \[Seite 714\]](#)

[Wertehilfe-Drucktaste \[Seite 759\]](#)

[Drucktasten \[Seite 711\]](#)

[Übersicht der Menüleisten \[Seite 679\]](#)

In addition to the guidelines listed under the individual control element keywords, this section provides general guidelines and notes on how to use control elements.

Selection Elements

The following table lists guidelines for use of the various selection elements. Which element is used depends on whether they support single or multiple selection (0...n), how many alternatives are available, and whether the alternatives are fixed and/or variable.

Selection	Alternative fixed/variable	Element	
single (1)	2...12 (*)	fixed (1 screen)	Radio buttons
multiple (0...12)	1...12	fixed (1 screen)	Checkboxes
single (1)	2...n	fixed (more than 1 screen)	Single-selection list
multiple (0...n)	1...n	fixed (more than 1 screen)	Multiple-selection list
single (1)	2...n	variable	Single-selection list
multiple (0...n)	1...n	variable	Multiple-selection list
*) 12 is just an example			

Pushbuttons

Since pushbuttons can occur at three locations in the R/3 System (application toolbar, standard toolbar, work area), you have to decide which of the standardized functions is to be found at which location. In addition, the following general guidelines apply, in particular to non-standardized functions:

- **Standard toolbar:** The [Standard Toolbar \[Seite 307\]](#) contains application-independent, general - so-called generic - functions such as *Save*, basic navigation functions (*Cancel*, *Exit*, *Back*, *Scroll*) and *Help*. In future, other generic functions are to be added to the standard toolbar.

Using Control Elements

- **Application toolbar:** The Application Toolbar (Control Element) contains application-specific functions which refer to a screen as a whole. This also includes navigation functions (except for the three basic functions *Cancel*, *Exit* and *Back*, which are part of the standard toolbar).
- **Pushbuttons with no fixed position:** The work area includes pushbuttons which refer to particular field contents, for example, pushbuttons for choosing a line or column of a tabular structure, or functions which sort and total certain fields.

Other examples are the Possible Entries Pushbutton (Control Element) and the More pushbutton on selection screens.

Finally, you can use Pushbutton with no fixed position to show or hide information (texts, graphics, animations).

On initial screens, pushbuttons are used in the work area to call the most important functions which can be performed on the chosen objects.

The system decides automatically if a task-defined function is displayed in the standard toolbar or application toolbar.

Please refer to Overview of Standard Functions for a list of the standardized functions including their position at the three locations.



Please refer to the individual control element keywords for more precise guidelines with regard to field positioning, field names, group boxes, and so on.

The graphic in [Graphical Overview of SAP GUI Terms \(German/English\) \[Seite 685\]](#) shows the most important SAP user interface terms in German and English.

Cross-References

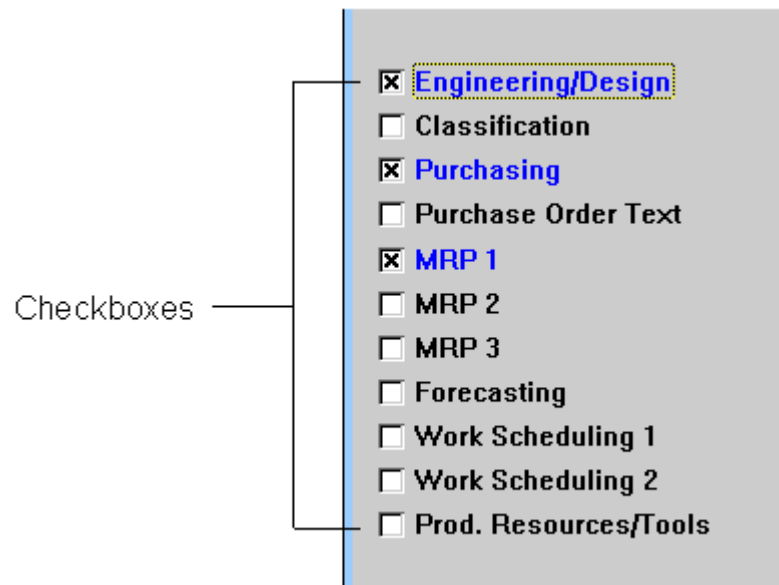
- Auswahlbuchstaben
- [Auswahltabellen und -listen \[Seite 362\]](#)
- [TabStrip \(Register\) \[Seite 314\]](#)
- [Access Characters \[Seite 184\]](#)
- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Checkboxes \(Control Element\) \[Seite 176\]](#)
- [Check Text \(Control Element\) \[Seite 300\]](#)
- [Command Field \[Seite 201\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Field Names \(Control Element\) \[Seite 324\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Function Key Menu \[Seite 278\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Group Box \(Control Element\) \[Seite 280\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Lists - Colors and Lines \[Seite 401\]](#)
- [Lists - Symbols and Icons \[Seite 432\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#)
- [Pushbuttons With No Fixed Position \(Control Element\) \[Seite 204\]](#)
- [Radio Buttons \(Control Element\) \[Seite 193\]](#)
- [Scroll Bars \(Control Element\) \[Seite 202\]](#)
- Single-Selection List
- [Standard Toolbar \[Seite 307\]](#)
- [Status Bar \[Seite 305\]](#)
- [Table Control \[Seite 340\]](#)

Checkboxes (Control Element)

Checkboxes (Control Element)

Checkboxes allow the user to choose items from a fixed number of alternatives.

Illustration



Checkboxes on a screen

Use

[Design Guidelines \[Seite 178\]](#)

[Technical Notes \[Seite 182\]](#)

[Cross-References \[Seite 183\]](#)

Use

Purpose

Checkboxes are used to choose **any number of options** from among a **limited number**.

You can select none, one, or more options from a group of checkboxes.

There may also be only one checkbox.

Appearance

The platform-specific design is used for checkboxes.

Operation/Functionality

Mouse

A mouse-click on the symbol or the text switches the select state of checkboxes between on and off (binary flag/toggle switch).

Keyboard

Checkboxes can be turned on or off using the keyboard by positioning the cursor on the element and pressing the space bar.

Naming Convention

Choose names which distinguish clearly between two different states or alternatives.

Design Guidelines

Design Guidelines

The Checkbox Refers to Adjacent Input Fields and Cannot be Separated From the Context

Checkboxes are to be left-aligned with their input fields.

The image shows two separate form panels. The left panel has a 'Group heading' and contains three rows. Each row has a 'Field label' followed by an input field. The first row's label is 'water' and the field contains 'aaa' with a checked checkbox. The second row's label is 'aaa' and the field contains 'aaa' with an unchecked checkbox. The third row's label is 'hkkk' and the field contains 'hkkk' with an unchecked checkbox. The right panel also has a 'Group heading' and contains three rows. Each row has a 'Field label' followed by an input field. The first row's label is 'text' and the field contains 'aa'. The second row's label is 'ghg' and the field contains 'ghg'. The third row's label is 'hjh' and the field contains 'hjh'. The fourth row's label is 'rtrt' and the field contains 'rtrt'.

Left alignment of checkboxes

They can also be placed to the right of the reference field or to the right of one of the reference fields, separated by blanks. If several reference fields are available, the checkbox should be aligned with the bottom reference field. Groups of checkboxes on the right should be justified.

The image shows a single form panel with a 'Group heading'. It contains four rows. The first row has a 'Field label' 'hjhjhj' followed by a 'Reference field' 'ghg' and a checked checkbox. The second row has a 'Field' 'gfgf' followed by an unchecked checkbox. The third row has a 'Reference field' 'fjhg' followed by an unchecked checkbox. The fourth row has a 'Field' 'hggh'.

Checkboxes to the right of the reference field

If there is not enough room to implement one of the above designs: the symbol to the right, aligned with the above input fields. Names are left-aligned with the above field names (For technical restrictions, refer to the Technical Notes).

Group heading	
Field label	gfgggf
Checkbox	<input type="checkbox"/>
Field	dfdfdf
Checkbox	<input checked="" type="checkbox"/>
Field	jkjk
Field	jjk

Field label	jk
Field	ghh
Field	fgg
Field	dddd

Checkboxes to the right of field names

The Checkbox Refers to Several Fields

Checkboxes which refer to several fields are to be left-aligned with the field names below the field group. You can insert a blank line between information groups (fields, radio buttons and checkboxes).

Group heading	
Field label	dfdfdf
Field	dfd
Field	
Field	gfggf
Field	qpqpqp
<input checked="" type="checkbox"/> Checkbox 1	
<input type="checkbox"/> Checkbox 2	

Field label	titit
Field	ttt
Field	
Field	rrrt

Checkboxes referring to several fields

The Checkboxes Represent a Separate Independent of Block Information

In this case, the checkboxes are to be placed as a separate group with or without surrounding box next to or below other blocks.

Design Guidelines

Group heading

Field label

Field

Field

Field

Field

Group heading

☒ Checkbox 1

☐ Checkbox 2

☒ **Checkbox 3**

☐ Checkbox 4

Checkboxes forming a separate information block

Sufficient Space Horizontally

If room is available horizontally but not vertically, you can position the checkboxes next to each other in one line. In this case, leave at least two blanks between the checkboxes.

You can write a name before the checkboxes, if required. Left-align the first checkbox with the input fields.

Group heading

Field label **Material description**

Field

Field

Label ☒ **Checkbox** ☐ **Checkbox** ☒ **Checkbox**

Horizontally arranged checkboxes with a name

If you do not use names, left-align the first checkbox with the field names.

Group heading		
Field label	fg	Material description
Field	gfh	
Field	hhgf	
<input checked="" type="checkbox"/> Checkbox	<input checked="" type="checkbox"/> Checkbox	<input type="checkbox"/> Checkbox

Horizontally arranged checkboxes without a name



Checkboxes in matrices are described under the keyword [Selection Tables and Lists](#) [\[Seite 362\]](#).

Technical Notes

Status Displays and Inactive Checkboxes

Two-value status displays (for example, "long text is available") and checkboxes that cannot be changed are displayed by inactive checkboxes until further notice.

Select

The select functionality is provided by the Basis Department.

Graphical checkboxes are provided by the new development workbench and defined in the Screen Painter. "Old" checkboxes must be changed manually with the Screen Painter.

Technical Restriction

If the keyword and the graphical symbol have identical names, the graphical element (the converted edit format) can currently only be positioned to the left of the keyword.

To position the symbol to the right of the keyword, they must thus have different names. In this case, however, the two elements are not connected, that is, you have to click directly on the symbol.

There are plans to lift these restrictions in the future. It will then be possible to arrange the graphical symbol to the left or right of the keyword and to create elements that are either connected or not. The naming restrictions will then also be removed.

Cross-References

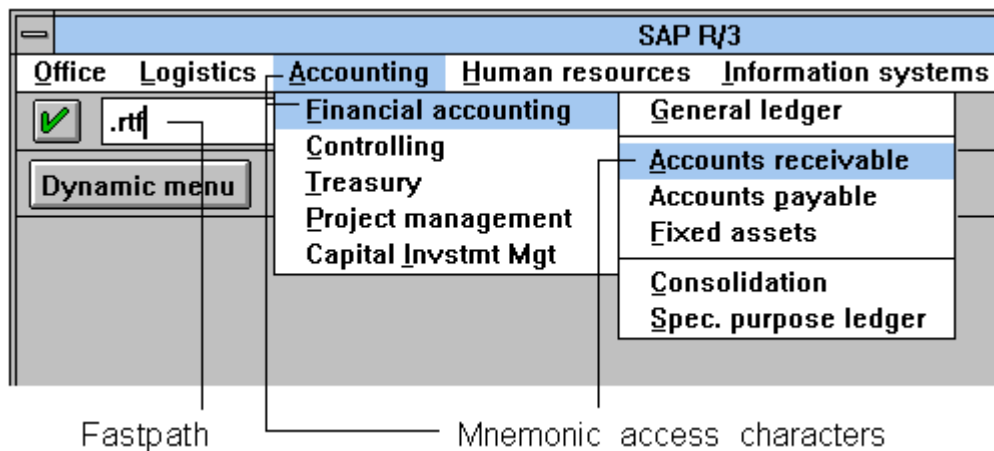
- [Control Elements - Overview \[Seite 168\]](#)
- [Radio Buttons \(Control Element\) \[Seite 193\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)

Access Characters

Access Characters

Access characters allow a menu option to be chosen with the keyboard. They can also be combined to form the fast path which is entered into the command field to call tasks or task functions directly.

Illustration



Access characters and fast path

[Use \[Seite 185\]](#)

[Access Character Overview \[Seite 186\]](#)

Pushbuttons with Icons on Initial Screens (Tools)

[Alphabetical List \[Seite 187\]](#)

[Classified List \[Seite 189\]](#)

[Technical Notes \[Seite 191\]](#)

[Cross-References \[Seite 192\]](#)

Use

Choosing Menu Options With The Keyboard

Access characters are used to choose a menu option with the keyboard by pressing the key combination `ALT + <access character>`. This opens the pull-down menu attached to the associated menu option or initiates the action defined for this menu option.

Fast Path

The access characters also combine to form a fast path for starting tasks or for initiating an action. To do this, you have to enter a period into the [Command Field \[Seite 201\]](#) followed by a string of access characters in the order of the sequence of menu options you would have to choose with the mouse in order to initiate the desired action.

Identification (Underscores)

Access characters are underlined.

Access Character Overview

Access Character Overview

The following overview lists the access characters for menu bar options and pull-down options in alphabetical order.

In addition, the lines are classified by menu bar or pull-down menu.

'*' Indicator

Menu options with defined access characters are flagged with '*' and have to be used. For options that are not flagged, the underlined character is a proposal. If you wish, you can also use a more appropriate letter.

Menu Options Not Listed

For menu options that are not contained in this list, you should use a letter which is characteristic for this item. Usually, this is the first letter or a letter which distinguishes different functions with the same first letter.



Access characters have to be **unique** in a menu bar and/or pull-down menu.

Consistent Use in the Application

Within an application, you should use access characters for the object classes and entries in the respective pull-down menus consistently.

Alphabetical List

	Append
*	B ack
*	C ancel
*	C hange
*	C hoose
*	C opy
*	Copy f rom
*	Cor r ect
*	Cr e ate
*	Cu t
*	D elete
	Delete line
*	D eselect all
*	Ex t ras
*	D isplay
*	E dit
*	Editor
*	E nvironment
*	Ex e cute
*	E xit
	F ind
*	Gener a te
*	G oto
*	H ader
*	H o ld
*	I nsert
*	L ong text
*	M ove
	N ew lines
*	N ext item
	O pen
*	Other i tem

Alphabetical List

	<u>O</u> ther <object>
*	<u>O</u> verview
	<u>P</u> aste
*	Post
*	<u>P</u> revious item
*	<u>P</u> rint
*	Reset
	<u>R</u> e-sort
*	<u>S</u> ave
*	<u>S</u> elect all
	Select <u>b</u> lock
*	<u>S</u> ettings
*	<u>S</u> ort
	<u>S</u> wap
	<u>T</u> ools
*	Undo
*	<u>U</u> tilities
	<u>V</u> iew

Classified List

Possible Options in the Menu Bar

*	<u>E</u> dit	*	<u>U</u> tilities
*	<u>G</u> oto		<u>T</u> ools
*	<u>E</u> xtras		<u>V</u> iew
*	<u>E</u> nvironment	*	<u>S</u> ystem
*	<u>S</u> ettings	*	<u>H</u> elp

Possible Options in the <Object> Pull-Down Menu

	<u>O</u> pen	*	<u>S</u> ave
*	<u>O</u> ther <object>	*	Post
*	<u>C</u> reate	*	<u>H</u> old
*	<u>C</u> hange	*	Gene <u>r</u> ate
*	<u>D</u> isplay	*	<u>P</u> rint
*	<u>E</u> xecute	*	<u>D</u> elete
*	Copy <u>f</u> rom	*	<u>E</u> xit

Possible Options in the Edit Pull-Down Menu

*	<u>S</u> elect all	*	<u>M</u> ove
	Select <u>b</u> lock		<u>S</u> wap
*	<u>D</u> eselect all		<u>R</u> e-sort
	<u>N</u> ew lines		<u>F</u> ind
	Appen <u>d</u>		De <u>l</u> ete line
*	<u>C</u> hoose	*	Corre <u>c</u> t
*	<u>C</u> u <u>t</u>	*	<u>S</u> ort
	<u>P</u> aste	*	Reset
*	<u>I</u> nsert	*	<u>C</u> ancel
*	<u>C</u> opy		

Possible Options in the GoTo Pull-Down Menu

*	<u>O</u> verview	*	<u>P</u> revious item
*	<u>H</u> ader	*	<u>O</u> ther item
*	<u>N</u> ext item	*	<u>B</u> ack

Classified List

Technical Notes

The access characters are defined in menu maintenance in the Menu Painter. You have to enter one access character in the *Fast path* column for every menu option. Make sure that all access characters are unique within a menu.

The Menu Painter makes access character suggestions.

Cross-References

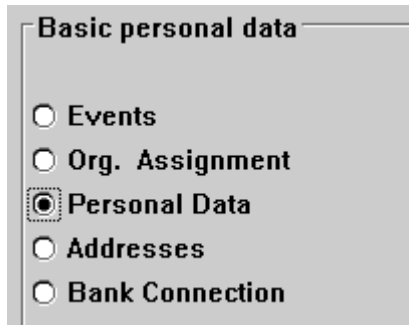
Cross-References

- [Command Field \[Seite 201\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Object-Oriented User Interface \[Seite 129\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Radio Buttons (Control Element)

Radio buttons allow the user to choose exactly one item from a list of several predefined alternatives.

Illustration



Radio buttons: one radio button is always selected

[Overview \[Seite 194\]](#)

[Design Guidelines \[Seite 195\]](#)

[Technical Notes \[Seite 199\]](#)

[Cross-References \[Seite 200\]](#)

Overview

Overview

Purpose

Radio buttons are used to choose **exactly one** option from a **limited number** of options.

You have to implement **at least** two radio buttons for one group.

One button must always be selected.

If the user should have the chance to choose **no** item from the offered options, you have to create a separate radio button with a "No selection" label or similar text which turns off the other contents-related options.

Appearance

The platform-specific appearance is used for radio buttons.

Use/Functionality

Mouse

A mouse-click on a non-selected item (symbol or name) activates it and deactivates the selected alternative of the group. A mouse-click on a selected alternative has no effect.

Keyboard

The active control indicator is positioned on an alternative which is not selected and the alternative is selected by pressing the space bar. The selected alternative of the same group is deactivated.

Design Guidelines

Related Input Fields

Radio buttons should be left-aligned with the input fields. You always have to use at least two radio buttons.

The image shows two side-by-side form panels illustrating radio button alignment. The left panel, titled 'Group heading', shows a 'Field name' with the value 'text' and three radio buttons labeled 'Radio button 1', 'Radio button 2', and 'Radio button 3'. Below this, a 'Field' with the value 'ghghgg' is shown, followed by two more radio buttons labeled 'Radio button 1' and 'Radio button 2'. At the bottom, another 'Field' with the value 'text' is shown. The right panel shows a 'Field name' with the value 'fgfg' and a series of six 'Field' labels, each followed by a text input field containing the values 'fgfg', 'hjhj', 'hghg', 'hghj', 'hgh', and 'tzt' respectively. The radio buttons in both panels are left-aligned with the input fields.

Left-aligned radio buttons

They can also be placed to the right of the reference field(s), separated by some blanks. If there are several reference fields, the radio button should be to the right of the bottom reference field. Several radio buttons of a group to the right of the fields should be left-aligned. You always have to use at least two radio buttons.

The image shows a single form panel titled 'Group heading'. It contains three sections. The first section has a 'Field name' with the value 'ghhg' and a 'Reference field' with the value 'jkjk', followed by three radio buttons labeled 'Radio button 1', 'Radio button 2', and 'Radio button 3'. The second section has a 'Field' with the value 'k1k1' and a 'Reference field' (empty), followed by two radio buttons labeled 'Radio button 1' and 'Radio button 2'. The third section has a 'Field' with the value 'jkjk' and no radio buttons. The radio buttons are left-aligned with the input fields and placed to the right of the reference fields.

Radio buttons to the right of the reference field

If there is not enough room to implement one of above described solutions, the following guideline applies: The symbol is to be placed to the right and left-aligned with the input fields above. Names are left-aligned with the field names above (for technical restrictions see the Technical Notes).

Design Guidelines

The image shows two side-by-side form panels. The left panel is titled 'Group heading' and contains a series of controls: a text field with 'dfdfdfdfdf', three radio buttons (the third is selected), a text field with 'ffd', two more radio buttons (the first is selected), and two text fields with 'dfg' and 'gfd'. The right panel contains a text field with 'dfff', followed by five text fields with 'ghg', 'hhg', 'ggh', 'fgg', and 'ddf', and a final text field with 'k1k1'.

Right radio buttons

Several Fields

Radio buttons which refer to several fields are to be left-aligned with the field names at the bottom of the field group. You can insert a blank line between groups of information units (fields, radio buttons, and checkboxes).

The image shows two side-by-side form panels. The left panel is titled 'Group heading' and contains a text field with 'ggfggf', followed by four text fields with 'ffgf', 'gggg', 'hgh', and 'hhgg'. Below these is a group of two radio buttons, 'Radio button 1' and 'Radio button 2', with the second one selected. The right panel contains a text field with 'gff', followed by three text fields with 'ggf', 'ffff', and an empty field.

Radio buttons which refer to several fields

Separate Independent Information Blocks

In this case, the radio buttons are to be placed as a separate group next to or below other blocks. Radio buttons are always related to each other and are therefore to be collated in a group box containing a group heading.

Group heading

Field name

Field Short text

Field

Field Short text

Field

Subgroup

☐ Radio button 1

☐ Radio button 2

☒ Radio button 3

☐ Radio button 4

Radio buttons which represent a separate information block

Horizontal Arrangement

If room is available horizontally but not vertically, you can position the radio buttons next to each other in one line. You have to leave at least two blanks between the radio buttons.

If necessary, you can position a name before the radio buttons. Then the first radio button is to be left-aligned with the input fields.

Group heading

Field name Material short description

Field

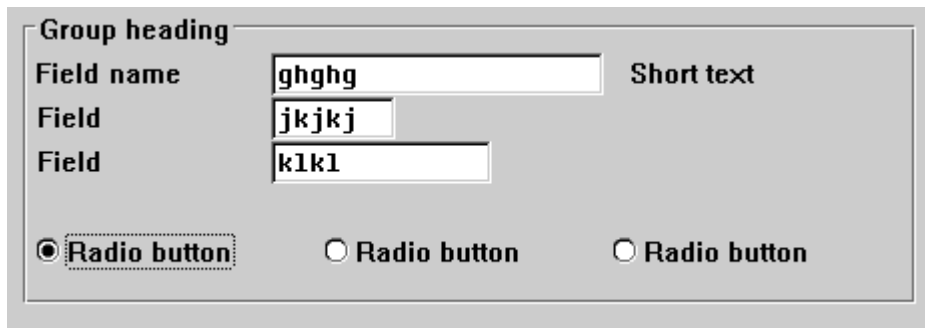
Field

Field name ☐ Radio button ☒ Radio button ☐ Radio button

Radio Buttons with horizontally arranged names

If you do not name the radio buttons, the first radio button is left-aligned with the field names.

Design Guidelines



The screenshot shows a SAP form layout within a grey border. At the top left is a 'Group heading' label. Below it are three text fields: the first is labeled 'Field name' and contains 'ghghg', the second is labeled 'Field' and contains 'jkjkj', and the third is labeled 'Field' and contains 'k1k1'. To the right of the first field is a 'Short text' label. At the bottom are three radio buttons, each followed by the text 'Radio button'. The first radio button is selected (filled circle).

Radio buttons without names



For more details on radio buttons in matrices, please refer to [Selection Tables and Lists \[Seite 362\]](#).

Technical Notes

Select

The Select functionality is made available by the Basis Department.

Groups

Related radio buttons are to be defined as a graphical group to guarantee that the system can initiate the radio buttons mutually.

Technical Restriction

If the keyword and the edit format have identical names, the graphical element (the converted edit format) can only be positioned to the left of the keyword.

Consequently, the keyword and edit format must have different names to position the symbol to the right. Both elements are, however, not connected with each other, that is, you have to click directly on the symbol.

In future, there are plans to lift these restrictions. It will then be possible to arrange the edit format to the left or right of the keyword and to create elements that are either connected or not. The naming restrictions will then also be cancelled.

Cross-References

Cross-References

- [Checkboxes \(Control Element\) \[Seite 176\]](#)
- [Control Elements - Overview \[Seite 168\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)

Command Field

The command field is a special input field in the R/3 System in which the user can enter typed commands, usually a fast path (see [Access Characters \[Seite 184\]](#)) or transaction codes.

Illustration



The command field in the standard toolbar

Appearance

Location

The command field is located in the [standard toolbar \[Seite 307\]](#) to the right of the `ENTER` pushbutton.

Structure

The command field is an input field allowing up to 10 characters which is implemented as a drop-down list. This list, called **command list**, contains the last 20 user commands. The user can choose an entry from this list as the current command. The list contains the commands entered in all open sessions.

Technical Note

The functionality of the command field is provided by the Basis Department.

Cross-References

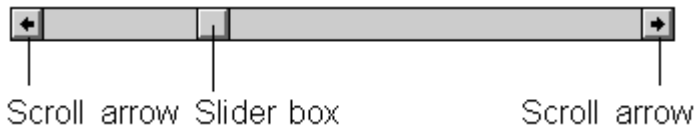
- [Access Characters \[Seite 184\]](#)
- [Control Elements - Overview \[Seite 168\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Scroll Bars (Control Element)

Scroll Bars (Control Element)

Scroll bars inform the user that there is more information available than displayed on the current screen. The user can view the hidden portions of the available information.

Illustration



Horizontal scroll bar

Purpose

Scroll bars can refer to an entire window as well as to parts of a screen such as tables or lists. They are to be used in the following cases:

- The size of the information exceeds the size of the window.
- A window is reduced so much that the displayed information does not fit in the window.

Appearance

Structure

Scroll bars consist of the following parts:

- the scroll arrows, which appear at each end of the scroll bar or which are displayed next to each other
- the scroll bar shaft represents the amount of information you can scroll through
- the slider box, which represents how far the current view of the information is from the top (for vertical scroll bars) or from the left edge (for horizontal scroll bars).

Design Guidelines

Scroll bars should extend over the entire height or width of the area to be scrolled.

Implementation

The platform-specific appearance is used for scroll bars.

Use/Functionality

A window can have vertical and horizontal scroll bars. They can only be operated with the mouse. You can scroll as follows:

- By clicking on the scroll arrows at the end of the scroll bar, you can scroll forwards or backwards **one line**.
- By clicking onto the free area in the scroll bar, you can scroll forwards or backwards **one screen**.

Scroll Bars (Control Element)

You can also scroll vertically screen by screen using the keys "Previous page" (F22 or PgUp), "Next page" (F23 or PgDn) or pushbuttons in the work area.

- By dragging the slider box, you can move quickly **anywhere** in the information. The slider box displays the current relative position. Depending on the platform, its size can be predefined or display the relative size of the visible section.



Other scroll options (first/last page, next/previous page) are offered with the function keys F21-F24 (or corresponding scroll keys on the keyboard) and the scroll arrows in the standard toolbar which are to be operated with the mouse.

Technical Notes

Scroll bars are made available by the Basis Department.

General

If the current window size is smaller than the screen size, horizontal or vertical scroll bars are activated automatically.

ABAP/4 Lists

Scroll bars are activated automatically in ABAP/4 lists. They are active if the list exceeds one screen page (depending on the window size).

Step Loops

In step loops, scroll bars can be activated manually. The following requirements are necessary to do this:

- an internal table is displayed in the step loop
- the ABAP/4 statement LOOP AT itab is used. Refer to the online help in the processing logic for LOOP.

In all other cases, usually no scroll bars appear.

Cross-References

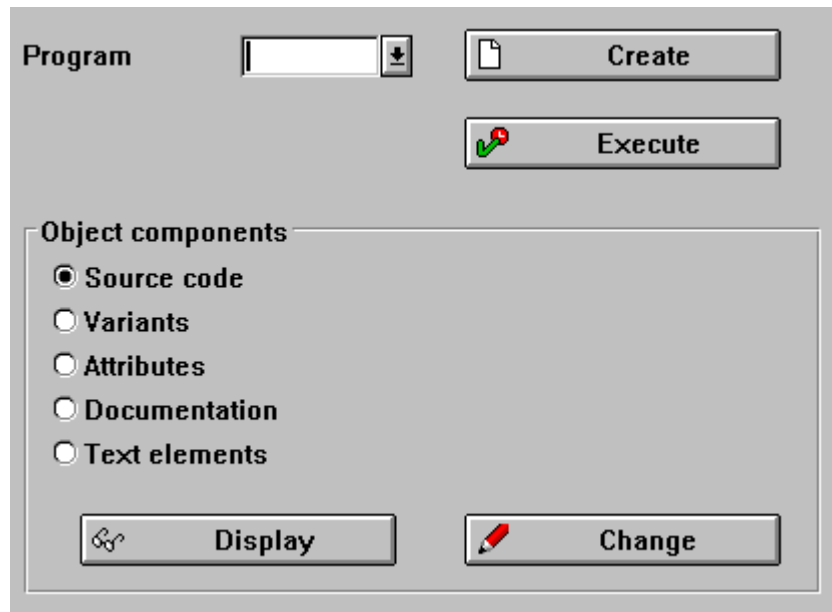
- [Control Elements - Overview \[Seite 168\]](#)
- [First Page, Last Page \(Function\) \[Seite 505\]](#)
- [Previous Page, Next Page \(Functions\) \[Seite 535\]](#)

Pushbuttons With No Fixed Position (Control Element)

Pushbuttons With No Fixed Position (Control Element)

Pushbuttons with no fixed position in the work area of a screen are used to initiate functions which are assigned to fields, tables or lists and/or their parts.

Illustration



Pushbuttons with no fixed position

[Use, Appearance and Control \[Seite 205\]](#)

[Arrangement on Data Screens \[Seite 207\]](#)

[Arrangement on Initial Screens \[Seite 212\]](#)

[Pushbuttons with Icons on Data Screens \[Seite 215\]](#)

[Pushbuttons with Icons on Initial Screens \(Tools\) \[Seite 220\]](#)

[Report Selection Screens \[Seite 222\]](#)

[Pushbuttons in Dialog Boxes \[Seite 223\]](#)

[Technical Notes \[Seite 225\]](#)

[Cross-References \[Seite 226\]](#)

Use, Appearance and Control

Use

Pushbuttons with no fixed position in the work area of a screen initiate functions which refer to individual fields, groups of fields, tables or lists an/or parts of them. For example, you can call detailed information on a field by pressing a pushbutton.

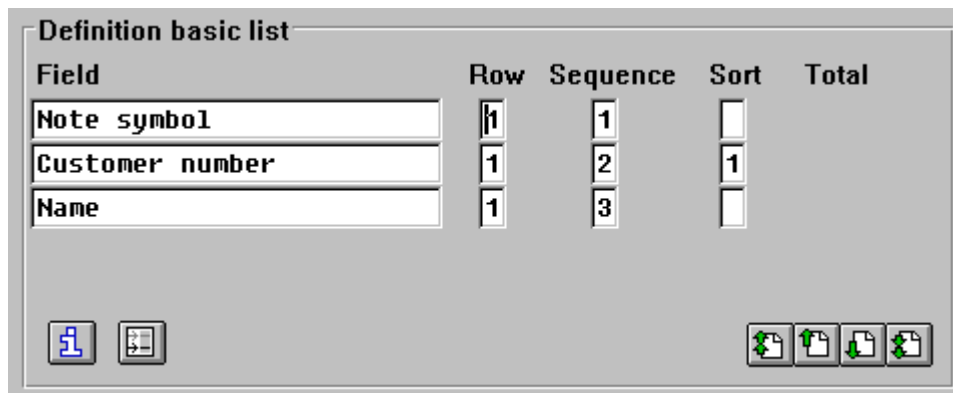
Further pushbuttons are predefined elements in the [Application Toolbar \(Control Element\) \[Seite 227\]](#) and in the [Standard Toolbar \[Seite 307\]](#).

Appearance

Pushbuttons may show text alone, text and icon, or solely an icon.

Pushbuttons with Icons

Icons need less room than icons. Therefore, use only pushbuttons with icons alone for the **functions describes below**. If the user positions the mouse cursor on a pushbutton, the system will - after a short delay - show its meaning in a small Quickinfo popup. Add text to icons only as an exception.



Example for pushbuttons with icons

Define the width of pushbuttons as follows:

- **Icon alone:** Set the width of the pushbutton to a value that the icon fits exactly on the pushbutton (as the Screen Painter suggests).
- **Icon and text:** The system places the icon always to the left of the text. Set the the width of the pushbutton in a way that its alignment and size fit to that of neighbouring pushbuttons and that there is sufficient room left for translation.

Realization

The platform-specific appearance is used for pushbuttons.

Use, Appearance and Control

Control

Mouse

A mouse-click initiates the function assigned to a pushbutton.

Keyboard

Pushbutton functions can be initiated with the keyboard by positioning the active control indicator (select) and pressing `ENTER` (activate).

Arrangement on Data Screens

Data Screens vs. Initial Screens

When using pushbuttons with no fixed position in primary windows, you have to distinguish between initial screens and data screens. For this reason, these screens are dealt with separately.

Using pushbuttons with no fixed position in dialog boxes is discussed afterwards.

Use

Pushbuttons with no fixed positions can be used on data screens for:

- calling further information.
Generally, a dialog box with additional information is sent. This can include further heading data, reference data or long texts, for example.
- executing immediate actions.
This can include *Print*, *Scroll* or *Delete*, for example.
- setting attributes.
The attributes can refer to the entire window (for example, hide/show heading) or to individual fields.

The screenshot displays a complex SAP data screen layout. At the top, there are three 'Header field' labels followed by input fields and a 'Description' label. To the right of these fields are two buttons: 'Hide' and a button with a green arrow and three dots. Below the header fields, there are two main sections. The left section is titled 'Group heading' and contains three 'Field' labels with input fields, a 'Description' label, and a 'Subgroup' section with radio buttons labeled 'None', 'Radio button 1', 'Radio button 2', and 'Radio button 3'. Below the radio buttons are two buttons labeled 'Action 1' and 'Action 2'. The right section is also titled 'Group heading' and contains a 'Field' label with an input field, a 'Reference' button, and two groups of checkboxes labeled 'Group 1' and 'Group 2'. Below these groups is a 'Field' label with an input field and a magnifying glass icon, and a button labeled 'Tests...'. At the bottom right, there is a 'Display:' label with two radio buttons labeled 'All' and 'Selection'.

Pushbuttons on screens for calling detailed information, actions and for setting screen attributes

Arrangement on Data Screens

Window- or Object-Related Pushbuttons

Pushbuttons which refer to the entire window or the full object are in the [Application Toolbar \(Control Element\) \[Seite 227\]](#) or [Standard Toolbar \[Seite 307\]](#).

Field-Related Pushbuttons

Pushbuttons which refer to fields are to be placed to the right of the field that they activate. In this case, the pushbuttons that are on top of one another should be aligned and, if necessary, should end at the right of the group box (see [Group Box \(Control Element\) \[Seite 280\]](#)). The buttons, however, should be close to the respective entry field so that they can be assigned instantly to the input fields.

Field-related pushbuttons should not appear in group boxes but should always be within the box of the group whose fields they activate.

The screenshot shows a data entry screen with a 'Material' field containing 'KF-BOTTLE' and a description field containing 'nail varnish bottle'. Below these is a section titled 'Inspection texts' with a 'Language' dropdown set to 'E' and a 'Long text' area containing the word 'bottle'. To the right of the 'Long text' area is a pushbutton with a document icon and a red arrow, which is field-related as it is positioned directly to the right of the text field it activates. A 'More long text' checkbox is also visible at the bottom right of the section.

Pushbutton on the long text as an example of a field-related pushbutton (middle right); pushbuttons which refer to the entire object are located at the upper window edge

Possible Entries Pushbutton

The Possible entries pushbuttons which are automatically displayed directly to the right of the field are a special case (see [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#), [Possible Entries \(Function\) \[Seite 499\]](#)).

The screenshot shows a data entry screen with fields for 'Material', 'Industry' (set to 'M'), and 'Material type'. To the right of the 'Material' field is a 'Possible Entries Pushbutton' (a small square button with a downward arrow) which is automatically displayed. Below these fields is a 'Reference' section with a 'Material' field and a pushbutton with a document icon and a red arrow, which is also field-related.

Possible Entries Pushbutton (Possible entries pushbutton, F4 functionality)

Group-Related Pushbuttons

Group-related pushbuttons are generally to be positioned at the bottom of the group. They should not be centered but left-aligned with the upper fields or group boxes.

A pushbutton in a [Group Box \(Control Element\) \[Seite 280\]](#) should always start in the first free column in the group box.

The screenshot shows a 'Group Box (Control Element)' with the title 'Object components'. Inside, there are five radio buttons for selection: 'Source code' (selected), 'Variants', 'Attributes', 'Documentation', and 'Text elements'. At the bottom of the group box, there are two pushbuttons: 'Display' (with a magnifying glass icon) and 'Change' (with a pencil icon).

Group-related pushbuttons

Table-Related Pushbuttons

Table-related pushbuttons are separated by one line from the table. They can be scroll functions, edit functions and functions for the positioning.

The screenshot shows a table with the heading 'Group heading'. The table has four columns labeled 'Column 1', 'Column 2', 'Column 3', and 'Column 4'. There are seven rows of data, each starting with a checkbox. Below the table, there are three pushbuttons: a magnifying glass icon, a green arrow pointing right, and a trash can icon.

Pushbuttons on a Table With Table Functions

Scroll Functions

The scroll buttons are right-aligned below the table. A possible existing page display should be positioned to the left next to it or in the line above. The scroll buttons should be within the box

Arrangement on Data Screens

enclosing the table if they refer exactly to the table. If they affect several tables, they should be outside the boxes. All pushbuttons should have a length of three characters. For scroll functions and their respective icons see [Pushbuttons with Icons on Data Screens \[Seite 215\]](#).



Detail Information				
Product group	CJ-HANAB1		Plant	0001
	M 12/1993	M 01/1994	M 02/1994	EH
Sales				
Production				
Warehouse stock				
Target stock level				
Days' supply				
Target days' supply				

Entry 2 of 14

⏪ ⏩ ⏴ ⏵

Horizontal scroll buttons

Conflicts With the Scroll Arrows of the Standard Toolbar

If there is only one step loop on the template, the vertical scroll buttons perform the function as the scroll arrows of the [Standard Toolbar \[Seite 307\]](#), however they do not interfere with one another. In this case, it is best not to include vertical scroll buttons.

If there are two or more step loops on the template, the general scroll functions (F21–F24) have to be switched off, since it is no longer clear to which step loop they refer. In this case, pushbuttons with no fixed position have to be added for vertical scrolling.

Size

All pushbuttons on a screen should generally be the same width. Adjacent pushbuttons of a group have to be the same width.

Minimum Width

Pushbuttons should be at least three characters wide. As for field names, the space required for the translation (see [Translation \[Seite 621\]](#)) is to be taken into account. For technical reasons, however, you always have to add a blank (to be entered as an underscore) because of the proportional font.

Maximum Width

Pushbuttons in the work area should not be wider than 30 characters.

Height

Pushbuttons automatically have the height of one line.

Arrangement on Initial Screens

General Notes

Initial screens are a special type of screen in a task. They are used to control the following task or evaluation. On an initial screen, the respective object is named and specified. Often, the screens to be displayed or the required processing method is chosen on initial screens. This controlling information and actions should be displayed directly in the work area of the initial screen because they display the central purpose of the initial screen.

On the other hand, these control functions are in the background on the following data screens of the task as compared to the task-specific information of the task. They are therefore not located in the work area of the data screen but in the menu and in the application toolbar.

Tool Tasks vs. Business Tasks

Tool tasks and business tasks differ with respect to the number of processing methods possible in a transaction. This difference is a result of the detailed user authorizations in the tasks. This differentiation should remain, but results in a slightly different design of the initial screens of the two task types.

Initial screens of **customizing transactions** should be designed following the guidelines specified for tool tasks.

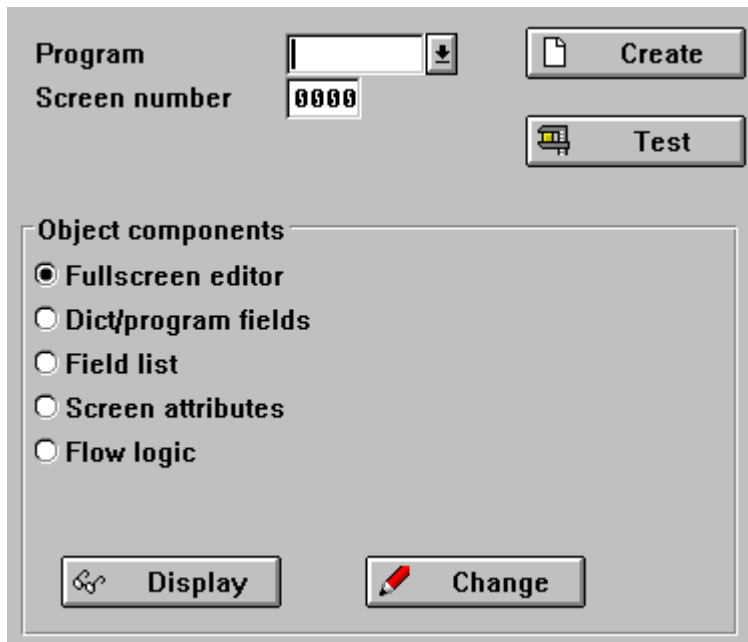
Arrangement

The object components and processing methods that can be freely accessed (tools or screens) can be positioned in the work area on initial screens of tool tasks.

The object components or processing methods displayed in the work area are **not** displayed in the application toolbar on the initial screen. The application toolbar can be equipped with other less important functions.

The object components or processing methods displayed in the work area, on the other hand, are always listed additionally in the menu so that they are also available on the data screens.

Arrangement on Initial Screens



Pushbuttons which refer to the entire object (above) and to object components with single selection (below)

Displaying

Object Components

- If only one object component can be selected at a time (single selection), the object components are to be designed as [Radio Buttons \(Control Element\) \[Seite 193\]](#) (see above).
- If several object components can be selected at the same time (multiple selection), the object components are to be displayed as [Checkboxes \(Control Element\) \[Seite 176\]](#).

Processing Methods

The processing methods possible within the selected transaction are only located in the work area in the tool tasks. There, they are always to be accessed with a pushbutton. If the processing methods refer to the entire object, the pushbuttons are to be positioned to the right of the input field of this object. If they refer to individual object components, they are to be arranged within or near the object components (see above).

If there is only one processing method, no *Edit* key is to be provided in the work area.

No more than ten pushbuttons should ever be positioned in the work area. Otherwise, transparency is lost and functions can no longer be quickly found.

If necessary, groups are to be separated from one another with a group box.

Initial screens should not contain much more elements than the initial screen shown in the following illustration.

Arrangement on Initial Screens

The screenshot displays a SAP initial screen with a light gray background. At the top left, the label "Program" is followed by a text input field and a small downward arrow icon. Below this, a section titled "Object components" is enclosed in a rounded rectangle. Inside this section, there is a list of radio buttons: "Status" (selected), "Status list", "Menu bars", "Menu list", "F key settings", "Function list", and "Title list". To the right of the "Status" radio button is a text input field and a "Test" button with a small icon. At the bottom of the "Object components" section, there are three buttons: "Display" with a magnifying glass icon, "Change" with a pencil icon, and "Create" with a document icon.

Initial screen with pushbuttons

Pushbuttons with Icons on Data Screens

On data screens, you can use icons for pushbuttons with no fixed position for a number of functions.

Generally, you should use a text on an iconized pushbutton only, if absolutely required.



"Find icon" and complete text *Find note*.

Horizontal and Vertical Scrolling

Icons for Horizontal Scroll Functions

You can use the following icons for horizontal scroll functions:

- Total left, Page left, Column left, Column right, Page right, Total right (see illustration below).

Appearance and Position

Only functions that are supported by the program are displayed as icons without a textual label (right-aligned below the step loop from left to right). Choose a width just large enough for the icon to fit. Leave one blank spacing between the pushbuttons.

<i>Total left</i>	<i>Page left</i>	<i>Column left</i>	<i>Column right</i>	<i>Page right</i>	<i>Total right</i>
TOTAL_ LEFT	PAGE_ LEFT	COLUMN_ LEFT	COLUMN_ RIGHT	PAGE_ RIGHT	TOTAL_ RIGHT

Icons for horizontal scroll functions (use the arrangement show)

Icons for Vertical Scroll Functions

Use icons for the vertical scroll functions only when there are several step loops on a screen or one step loop on a long screen:

- First page, Previous page, Next page, Last page (that is, F21 through F24)

Appearance and Position

The pushbuttons should be positioned to the right of the step loop (on top of each other). As an exception you can position them right below the step loop with one blank spacing. Possible pushbuttons for horizontal scroll functions should be arranged as a separate group (four characters spacing) to the left of the vertical scroll buttons.

<i>First page (F21)</i>	<i>Previous page (F22)</i>	<i>Next page (F23)</i>	<i>Last page (F24)</i>
FIRST_PAGE	PREVIOUS_PAGE	NEXT_PAGE	LAST_PAGE

Pushbuttons with Icons on Data Screens

Icons for vertical scroll functions (arranged as shown or vertically from left to right)



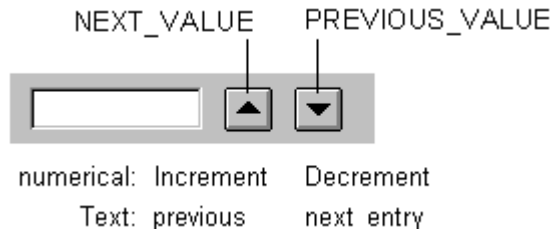
As soon as table controls will be available as a new graphical element these iconized functions will be superfluous. Table controls have separate horizontal and vertical scroll bars.

Spin Buttons

A spin button is a control element for the sequential display and/or input of mutually exclusive alternatives. Use spin buttons to display or enter alternatives with a logical ascending order, for example, the months of the year.

Appearance

You display the functionality of a spin button using an input field and one of the following icons (see illustration) always without a textual label:



Spin Button example

Two pushbuttons to the right of an input field, one character spacing (spin button) with the meaning (from left to right): "Increment" and "Decrement" (numeric values) and "Previous entry" and "Next entry" (texts).

Use with Alphanumeric and Numeric Values

When choosing alphabetical entries (for example, month, weekday) the upwards pointing arrow means *Previous entry* and the downwards pointing arrow *Next entry*.

With numeric values, (for example, the rotation angle of an object, time) the upwards pointing arrow means *Next value (Increment)*, the downwards pointing arrow *Previous value (Decrement)*.

At the beginning or the end of the list the system accesses cyclically the other end of the list.

Word Processing Functions

The following functions are usually displayed as minimum width pushbuttons without a textual label. You should position them in the work area in a way that makes the intended reference obvious (usually to the right of an object). You should use text only if absolutely required.

Example: If the address icon for different types of addresses is used (shipping address, order address etc.), you have to complete the icon with a corresponding text.

Pushbuttons with Icons on Data Screens

(Change) Text (word processing)







- Following the *Change text* icon (default, if *Create*, *Change* and *Display* are not distinguished) an intermediary screen can come up (for example, dialog box for choosing a language) from where the user can call the SAPscript word processing. The user can also start SAPscript directly. Formatted texts can be entered. This can also be an OLE object (WinWord, Wordperfect etc.). This function refers to the entire object (teilweise in Release 2.1 Bezeichnung "Long text", for example, with transaction "Change cost center").
- Extension: Icons with the meaning *Display text* and/or *Create text* are available to set the three text icons at runtime. A separate status display that text is already available, is being created or can only be displayed, is superfluous.

Scheme:

Text	Edit Mode	Display Mode
no text	Create text, active	Create text, inactive
no text, protected	Create text, inactive	---
text exists	Change text, active	Display text, active
text exists, protected	Display text, active	---

(Change) Note, Comments

- entering comments with reference to a particular action (object component, for example, order item) or a particular line of a report. Typically, these texts are entered unformatted.
- Extension as with text (word processing): Create note and/or comment and/or Display note and/or comment

	CHANGE_TEXT	(Change) text		CREATE_TEXT	Create text
				DISPLAY_TEXT	Display text
	ANNOTATION	(Change) annotation		CREATE_NOTE	Create note
				DISPLAY_NOTE	Display note

Icons for text/word processing and annotation functions











Other Services

For other services you can use the following functions:

- Table calculation:** Calling for example the unit calculation from a previous application, for example, "Change part"
- (Business/Statistics) graphics:** Calling the business graphics from a window in which displays a list with data
- Variants:** Calling a dialog box, in which layout variants can be specified (for exdample, in the Report Writer or in Logistics-Controlling) or display variants can be selected (for example in the line item display)

Pushbuttons with Icons on Data Screens

- **Info:** Information text from application (that is, created by SAP or generated by the application or a data processing administrator). Texts can refer to the application (for example, reports in CO) or to individual columns in a report (EIS). They describe specific aspects of an application and cannot be changed by the user.
- **Short Message:** May be used in mail applications
- **Address:** Calling the address dialog box
- **(Set) status:** Calling a window to set the status for an object (for example, entering release infos in the IMG)
- **Tools:** Calling Tools inside or outside the R/3 system
- **Export/Import:** Saving files on a PC/loading files from a PC

	CALCULATION	<i>Spreadsheet, Calculation</i>		GRAPHICS	<i>Graphics</i>
	SHORT_MESSAGE	<i>Short message</i>		ADDRESS	<i>Address</i>
	INFORMATION	<i>Information</i>		SET_STATE	<i>Set state</i>
	VARIANTS	<i>Variants</i>		TOOLS	<i>Tools</i>
	EXPORT	<i>Export, Download</i>		IMPORT	<i>Import, Upload</i>

















Icons for further functions

Functions for Table Structures

You can use icons for the following table structure functions (see illustration):

- Choose detail (F2) (activates window with a detail screen containing the line contents)
- Insert row in table structure
- Delete row in table structure
- Sort (up) (default and/or natural sort order; from A to Z and/or 1 to 100)
- Sort down (reverse sort order, for example from Z to A)
- Position (on a particular line)
- Replace values in a column
- Search for a value in a column
- Sum (calculation sub-totals)
- Move (from column to line)
- Expand (particular columns)
- Compress (after a particular column)
- Selection functions: Select/Deselect all, Select block

Pushbuttons with Icons on Data Screens

	MOVE	<i>Move</i>		REPLACE	<i>Replace</i>
	INSERT_ROW	<i>Insert row</i>		DELETE_ROW	<i>Delete row</i>
	SEARCH	<i>Search</i>		SEARCH_NEXT	<i>Search next</i>
	SORT_UP	<i>Sort up</i>		SORT_DOWN	<i>Sort down</i>
	SUM	<i>Totals</i>		SELECT_DETAIL	<i>Select detail</i>
	EXPAND	<i>Expand</i>		POSITION	<i>Position</i>
	COLLAPSE	<i>Collapse</i>		SELECT_BLOCK	<i>Select Block</i>
	SELECT_ALL	<i>Select all</i>		DESELECT_ALL	<i>Deselect all</i>

Icons for table structures

Appearance and Position

Iconized pushbuttons are usually positioned left aligned with one character spacing below the table control and/or the step loop. You can use icons with or without textual labels. Icons should have text if there is enough room below the step loop and/or if the text explains the function of the icon. If there is not enough room, you can also position the pushbuttons to the right of the table control and/or the step loop (aligned with the bottom horizontal border of the table control).

Pushbuttons with Icons on Initial Screens (Tools)

Pushbuttons with Icons on Initial Screens (Tools)

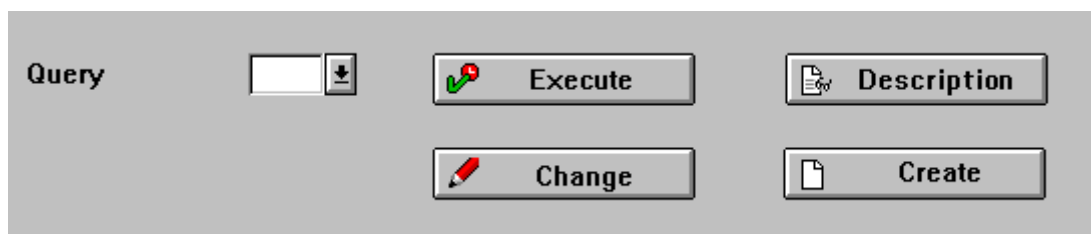
On initial screens, the text of the following generic pushbuttons (see illustration) is assigned an icon:

Create, Change, Display, Display <-> Change, Execute, Replace, Copy, Delete, Find.



This applies basically to initial screens of Basis applications, where above mentioned functions are implemented as pushbuttons with no fixed position.

The illustration shows how the pushbuttons are to be arranged:



Initial Screen example

Appearance and Width of Pushbuttons

If there is enough room on initial screens, these functions are always displayed as icons with a textual label unless there is not enough room.

	CREATE	<i>Create</i>		REPLACE	<i>Replace</i>
	CHANGE	<i>Change</i>		COPY_OBJECT	<i>Copy<object></i>
	DISPLAY	<i>Display</i>		DELETE	<i>Delete</i>
	TOGGLE_DISPLAY_CHANGE	<i>Display <-> Change</i>		EXECUTE_OBJECT	<i>Execute</i>
	SEARCH	<i>Find</i>		SEARCH_NEXT	<i>Find next</i>

Icons for general functions on initial screens











Application-specific Functions

For the Workbench you can use icons for the following functions:

Activate, Generate, Check, Test; List icons, Navigation stack, Where-used list








	ACTIVATE	<i>Activate</i>		ICON_LIST	<i>Icon list</i>
--	----------	-----------------	--	-----------	------------------

Pushbuttons with Icons on Initial Screens (Tools)

	GENERATE	<i>Generate</i>		STACK	<i>Stack</i>
	CHECK	<i>Check</i>		REFERENCE_ LIST	<i>Reference list</i>
	TEST	<i>Test</i>		TRANSPORT	<i>Transport</i>
	COMPARE	<i>Compare</i>		TOOLS	<i>Tools</i>
	RENAME	<i>Rename</i>		BREAKPOINT	<i>Breakpoint</i>

Special functions (Workbench)

You can use the following pushbuttons for representing networks and hierarchies:





















	PREVIOUS_ NODE	<i>Previous node</i>		NEXT_NODE	<i>Next node</i>
	PREVIOUS_ HIER_LEVEL	<i>Previous hierarchy level</i>		NEXT_HIER_ LEVEL	<i>Next hierarchy level</i>
	TREE	<i>Tree</i>		INSERT_ RELATION	<i>Insert relation</i>
	FINAL_DATE	<i>Final date</i>			

Special icons for networks

Report Selection Screens

Report Selection Screens

The following icons are available to be used especially on report selection screens. In addition, you can use these icons in the *Selection options* dialog box as field names with green, yellow or red background, depending on the context.

	DISPLAY_ MORE	<i>Display more</i>		ENTER_MORE	<i>Enter more</i>
	NOT_EQUAL_ GREEN	<i>Not equal</i>		NOT_EQUAL_RED	<i>Not equal</i>
	GREATER_ GREEN	<i>Greater</i>		GREATER_RED	<i>Greater</i>
	LESS_GREEN	<i>Less</i>		LESS_RED	<i>Less</i>
	GREATER_ EQUAL_GREEN	<i>Greater or equal</i>		GREATER_ EQUAL_RED	<i>Greater or equal</i>
	LESS_EQUAL_ GREEN	<i>Less or equal</i>		LESS_EQUAL_ RED	<i>Less or equal</i>
	INTERVAL_ EXCLUDE_ GREEN	<i>Include interval</i>		INTERVAL_ INCLUDE_RED	<i>Include interval</i>
	INTERVAL_ EXCLUDE_ GREEN	<i>Exclude interval</i>		INTERVAL_ EXCLUDE_RED	<i>Exclude interval</i>
	PATTERN_ INCLUDE_ GREEN	<i>Include pattern</i>		PATTERN_ INCLUDE_RED	<i>Include pattern</i>
	PATTERN_ EXCLUDE_ GREEN	<i>Exclude pattern</i>		PATTERN_ EXCLUDE_RED	<i>Exclude pattern</i>

Icons for ABAP/4 report selection screens (green = select, red = exclude from selection)

You can use these icons for pushbuttons and as a replacement for field names which appear flat on the template.

Pushbuttons in Dialog Boxes

The guidelines specified for data screens also apply to modal dialog boxes:

- Functions which refer to the entire dialog box are located in the application toolbar and defined in the Menu Painter.
- Functions which refer to individual objects of the dialog box are to be placed directly at the object as pushbuttons with no fixed position in the Screen Painter. They should be arranged horizontally.

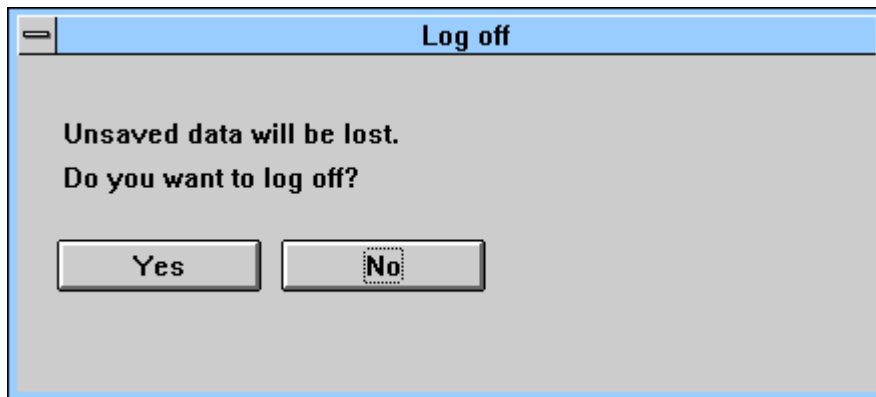
No.	Report	Type	Name
001	8A20	RGRP	Plan/Actual Comparison
002	8A29	RGRP	P/A Comp. of Profit Center Group
003	AS15	RGRP	P/A Comp. of Profit Center Group
004			

Functions which refer to the entire dialog box are in the application toolbar. Functions which refer to individual objects of the dialog box are to be implemented as pushbuttons with no fixed position.

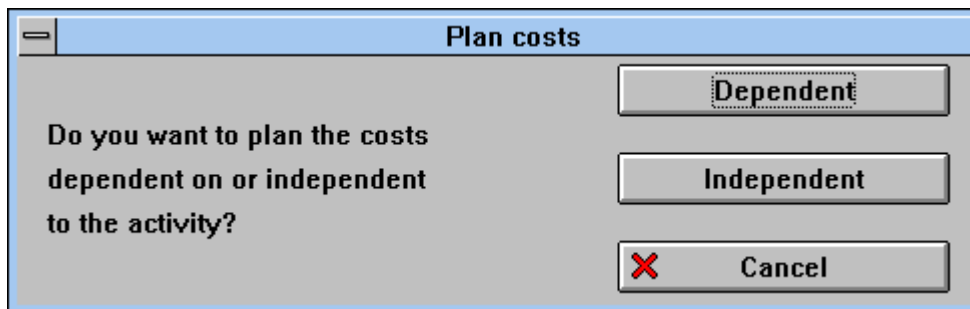
No Application Toolbar

For the SAP standard dialog boxes of the dialog modules "POPUP_TO_..." (for example, confirm prompt), the application toolbar is not to be used. The following exception rule applies to these dialog boxes: The pushbuttons are not in the application toolbar and can be arranged vertically at the right border.

Pushbuttons in Dialog Boxes



In SAP standard dialog boxes, the application toolbar is not to be used; the pushbuttons are arranged horizontally at the bottom.



In SAP standard dialog boxes, the application toolbar is not to be used; the pushbuttons are arranged vertically to the right.



For information on how to use icons on pushbuttons in dialog boxes, please refer to [Application Toolbar \(Control Element\) \[Seite 227\]](#).

Technical Notes

Definition

Unlike menu function, pushbuttons with no fixed position are defined in the Screen Painter.

Icons

Pushbutton icons are defined as one line and always at least two characters wide. Leave one blank character to the right and the left of the pushbutton.

- **Static pushbuttons**

Static icons on pushbuttons are defined in the Screen Painter as static texts on pushbuttons.

- **Dynamic pushbuttons**

Dynamic icons on pushbuttons are defined in the Screen Painter as dynamic texts on pushbuttons. They are assigned names and icon descriptions at runtime. This means, you can specify the icon only at runtime.

Cross-References

Cross-References

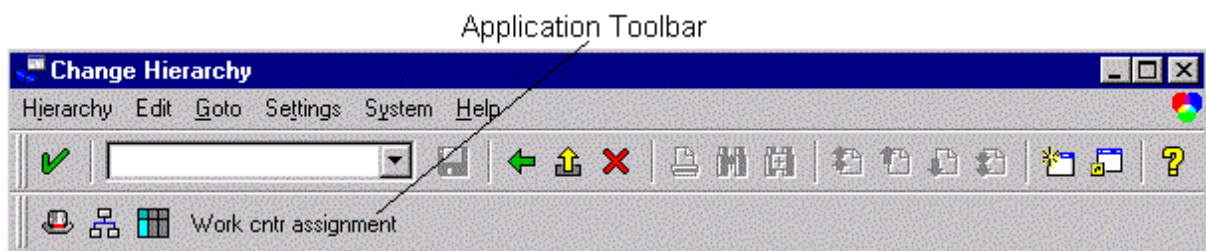
- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Control Elements - Overview \[Seite 168\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)
- [Standard Toolbar \[Seite 307\]](#)
- [Table Control \[Seite 340\]](#)

Application Toolbar (Control Element)

The following guidelines specify the order in which pushbuttons are to be arranged in the primary window application toolbar.

The design of the application toolbar in dialog boxes is described in the section [Modal Dialog Box \[Seite 65\]](#).

Illustration



The application toolbar

[Use and Position \[Seite 228\]](#)

[Design Guidelines \[Seite 229\]](#)

[Icons in the Application Toolbar \[Seite 231\]](#)

[Icons on Pushbuttons in Dialog Boxes \[Seite 234\]](#)

[Technical Notes \[Seite 236\]](#)

[Cross-References \[Seite 237\]](#)

Use and Position

Use and Position

The application toolbar offers quick access to application-specific functions for mouse users.

Application Toolbar vs. Standard Toolbar

The application toolbar contains the application-specific functions. The standard toolbar contains cross-application functions, so-called generic functions.

Position

The application toolbar is located below the [Standard Toolbar \[Seite 307\]](#), directly above the primary window work area.

Design Guidelines

[Sichern ohne Prüfen \[Seite 526\]](#)

Display All Pushbuttons

Try to display all the pushbuttons that are important for a task (approx. 6 - 7). Use appropriate abbreviations, to fit them all in the toolbar (see [Abbreviations \[Seite 593\]](#)).



Since some general functions are implemented in the standard toolbar, you can accommodate more functions in the application toolbar. Standard toolbar functions are not discussed further in this section (see [Standard Toolbar \[Seite 307\]](#)).

Consistent Terminology

Always use the same terms and abbreviations for the same function in your task (see [Formulations \[Seite 601\]](#) and [Functions - Overview \[Seite 451\]](#)).

Standard Functions

You can display those standard functions F2=[Choose \[Seite 487\]](#), F13=[Print \[Seite 496\]](#) and F14=[Delete \[Seite 515\]](#) which are active.

You can also display other standard functions in the application toolbar. This includes the [Hold \[Seite 519\]](#), Save without check and [Sort \[Seite 527\]](#) functions which are assigned to general function keys.

Do Not Display

the F9=SELECT (if defined with [Select/Deselect \(Function\) \[Seite 516\]](#)) and F10=MENU BAR pushbuttons (see [Menu Bar \(Function\) \[Seite 518\]](#)).

Justification: It does not make sense to define pushbuttons for these functions because it is easier to click directly on the specified object.

Self-Defined Function Keys

The self-defined function keys (always define F5-F8 first) should be displayed as pushbuttons if possible. At least the most frequently used functions should always be implemented as pushbuttons.

Justification: These pushbuttons cover specific, self-defined functions, which are difficult to learn and to remember.

Additional Navigation Functions

If the navigation keys are used for moving between the task hierarchy levels (F17-F20), they can be displayed.

Grouping Pushbuttons

Group pushbuttons in the following order:

Design Guidelines

1. the displayed active **standard functions**
2. one of the hierarchical navigation functions F17=[Other <object> \(Function\) \[Seite 474\]](#) or F17=[Open \(Function\) \[Seite 512\]](#), F18=[Previous <Object component> \[Seite 538\]](#), F19=[Next <object component> \[Seite 538\]](#) AND F20=[Other <object component> \(Function\) \[Seite 476\]](#), if available
3. **self-defined** functions

Within these three groups, the pushbuttons are to be arranged in ascending order of the corresponding function keys.



In dialogboxes, [Cancel \[Seite 470\]](#) is always to be the rightmost pushbutton.

Icons in the Application Toolbar

If an icon is available for a function, you should include it in the application toolbar either with or without text label.



Functions with icons in the application toolbar in the primary window

Icons without textual label

The following functions use icons without a textual label:

- Execute, Find, Replace, Copy, Delete, Sort, Total
- On detail screens: Previous entry (item, record), Next entry (item, record)
- With screen sequences: Previous <object component; screen>, Next <object component; screen> (see below) (Icon without textual label)
- With reports or hierarchies: Expand, Collapse
- With editing objects: Other <object>, Create, Change, Display, Display <-> Change, if located in the application toolbar.

	EXECUTE_OBJECT	<i>Execute</i>		TOTAL_LEFT	<i>First item/position</i>
	REPLACE_OBJECT	<i>Replace</i>		COLUMN_LEFT	<i>Previous item/Position</i>
	COPY_OBJECT	<i>Copy</i>		COLUMN_RIGHT	<i>Next item/position</i>
	DELETE	<i>Delete</i>		TOTAL_RIGHT	<i>Last item/position</i>
	SORT_UP	<i>Sort (up)</i>		SORT_DOWN	<i>Sort (down)</i>
	SEARCH	<i>Find</i>		SEARCH_NEXT	<i>Find next</i>
	SUM	<i>Total</i>		OTHER_OBJECT	<i>Other object</i>
	PREVIOUS_OBJECT	<i>Previous object component, Previous object</i>		NEXT_OBJECT	<i>Next object component, Next object</i>
	COMPARE	<i>Compare</i>		FILTER	<i>Filter</i>

Icons in the Application Toolbar

	RENAME	<i>Rename</i>		TRANSPORT	<i>Tranport</i>
	EXPORT	<i>Export, Download</i>		IMPORT	<i>Import, Upload</i>

Icons for application toolbar functions

	NEXT_VALUE	<i>Previous item, next value</i>		PREVIOUS_VALUE	<i>Next item, Previous value</i>
--	------------	----------------------------------	--	----------------	----------------------------------

Icons used in dialog boxes with reference to list items (spin button icons)

	EXPAND	<i>Expand</i>		COLLAPSE	<i>Collapse</i>
	NEXT_HIER_LEVEL	<i>Next hierarchy level</i>		PREVIOUS_HIER_LEVEL	<i>Previous hierarchy level</i>

Icons for functions on hierarchies

Icons with Textual Label

If there is enough room, you can include icons with a textual label for special functions such as *Refresh*, *Convert*, *Skip*, or *Retrieve*. You should add text to the icon, when the icon only classifies the class and a description is required to understand the full meaning of the function.

	REFRESH	<i>Refresh</i>		HEADER	<i>Header data, Basic data</i>
	SKIP	<i>Skip</i>		OVERVIEW	<i>Overview data</i>
	CONVERT	<i>Convert</i>		SELECT_DETAIL	<i>Choose (detail)</i>
	RETRIEVE	<i>Retrieve</i>		LIST	<i>List</i>

Icons for application toolbar functions that are usually accompanied by a text label



Check if the pushbutton can be positioned next to the object in the work area. In the application toolbar include icons for application-wide functions.

Sequence

Follow the following convention:

- **Principle 1:** Position application-wide functions at the left in the application toolbar.
- **Principle 2:** Icons without text label should always appear to the left of icons or pushbuttons with text labels as long as this is not in conflict with Principle 1.

Icons in the Application Toolbar

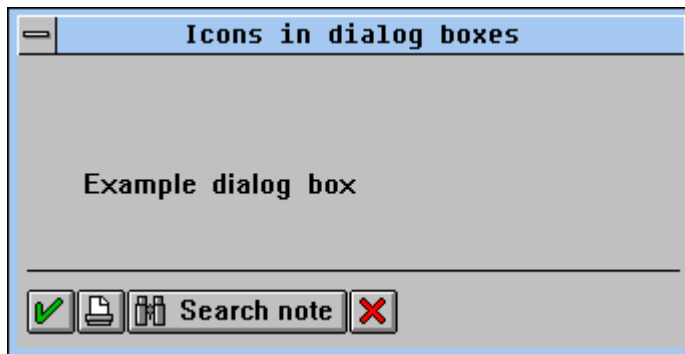
Position the icons for the functions *Previous <object component>* (if available) and *Next <object component>* (if available) always at the beginning of the application toolbar. An aesthetically pleasing overall impression decides if the icons should have a text label or not.

Icons on Pushbuttons in Dialog Boxes

Icons on Pushbuttons in Dialog Boxes

With very few exceptions, the functions of the `ENTER` key are always represented as icons. This also applies to the *Cancel* function.

Illustration



Icons in the dialog box

Appearance

The Default Key (`ENTER`)

In general, the `ENTER` key is only assigned the checkmark icon for the functions *OK*, *Continue*, *Close* or *Execute*.

When the *Choose* and/or the *Configure* functions are assigned to the `ENTER` key you should use the icon with a text label, for example "Icon_OKAY" plus text "Choose". For specific functions on the `ENTER` key such as *Delete*, *Check*, *Refresh* or *Find* use the checkmark icons plus text.

Cancel




Cancel (the last button in the application tool bar) is only assigned an icon.

Other Functions

If there is enough room, add text to the icon. Otherwise use an icon without a text label.

Exceptions

If the dialog box displays text only (for example, confirmation prompts), you should use an icon with a textual label. Messages dialogs do not have icons on the pushbuttons.

	OKAY	<i>Execute, Continue</i>
	CANCEL	<i>Cancel</i>
	PRINT	<i>Print</i>

Specific icons for functions in dialog boxes

Icons on Pushbuttons in Dialog Boxes**Sequence**

As specified above: ENTER button - dialog box specific pushbuttons - *Cancel*

Pushbuttons which have an icon but no text should be positioned directly after the ENTER button. All other buttons or icons follow.

Technical Notes

Pushbutton texts

- are defined in the Menu Painter
- have a maximum length of 20 characters
- must be at least five characters long (if the function name is shorter than five characters, e.g. "End", blanks are added automatically to the text to a length of five characters)
- should be one or two characters longer than the actual text (using underscores as in the Screen Painter) to simplify translation
- are concatenated dynamically depending on their length
- are centered within their fields
- must have three periods (...) after their name if the function initiates a dialog box.

Icons

You can define icons to be used in the application toolbar in the Menu Painter.

Cross-References

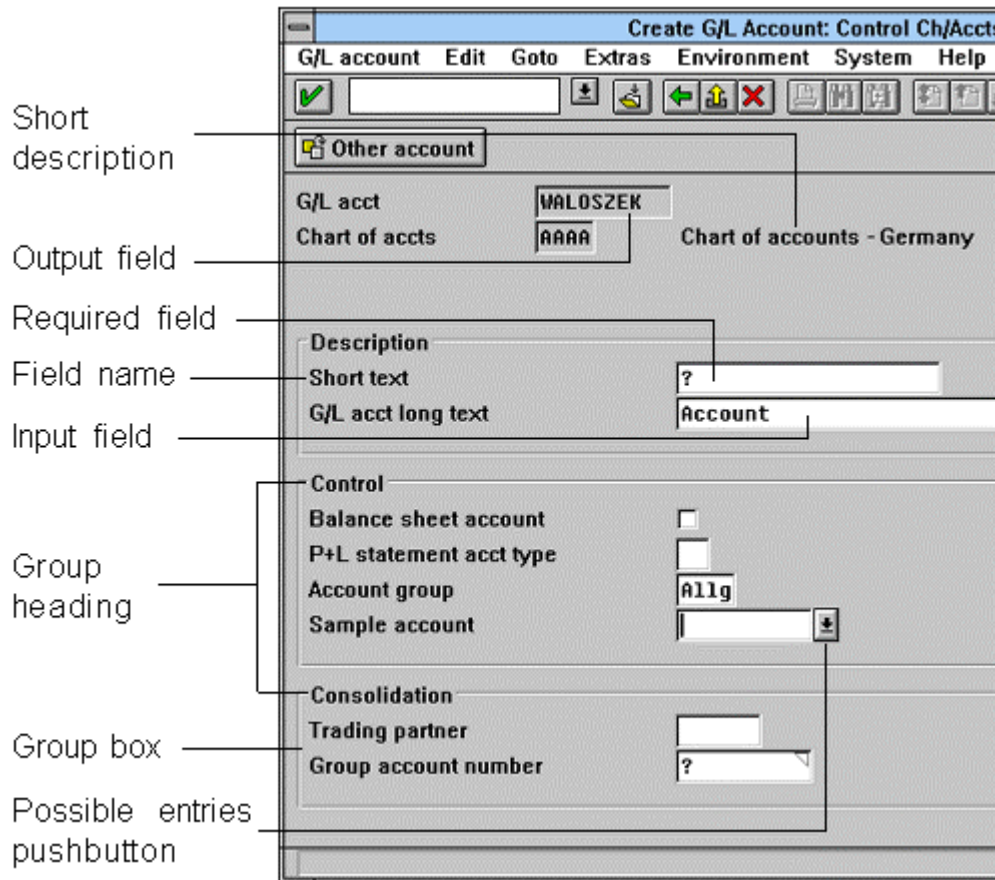
- [Abbreviations \[Seite 593\]](#)
- [Control Elements - Overview \[Seite 168\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)
- [Formulations \[Seite 601\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Modal Dialog Box \[Seite 65\]](#)
- [Pushbuttons With No Fixed Position \(Control Element\) \[Seite 204\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Input/Output Fields (Control Elements)

Input/Output Fields (Control Elements)

Input/output fields are fields in the work area of a window into which the user can enter information or in which the system displays already known data.

Illustration



Screen with recessed input/output fields: input fields have a highlighted background, recessed output fields have the same color as the window background.

Required fields are indicated by a question mark. Check texts are input/output fields and are flat on the template.

A Possible entries pushbutton (non-permanent Possible entries pushbutton) is displayed for fields which support the Possible entries function, if the active control indicator is positioned on the field.

Matchcode fields under Windows have a colored triangle in the upper right corner.

Fields can be collated in group boxes which are identified by a group heading.

[Purpose \[Seite 240\]](#)

[Appearance \[Seite 241\]](#)

Input/Output Fields (Control Elements)

[Design Guidelines \[Seite 243\]](#)

[Scrollable Input/Output Fields \[Seite 247\]](#)

[Technical Notes \[Seite 249\]](#)

[Overview of the Field Attributes \[Seite 250\]](#)

[Cross-References \[Seite 252\]](#)

Purpose

Purpose

Input/output fields (I/O fields) are fields in the work area of a window in which the user can enter information or in which the system displays already known data.

Input/output fields can be ready for input or not ready for input.

Read-only output fields are only used for displaying information. In the R/3 System, they are used as [Field Names \(Control Element\) \[Seite 324\]](#), to display table or column headings, and explanatory texts (see [Check Text \(Control Element\) \[Seite 300\]](#)).

Appearance

Display

Depending on their function, input/output fields look differently. The corresponding flags are to be set for this in the Screen Painter (see Technical Notes).

The display of the input fields depends on the respective platform and uses the system resources provided for this purpose.

CUA, Motif, Windows

Input fields are displayed three-dimensionally, recessed in the surface of the template.

Apple

Input fields are displayed two-dimensionally and have an L-shaped border.

Color

Input/output fields have the color of the platform-specific system resources.

The color of the 3D shadow or the 2D border depends on the system resources; if no resources are available, black is used.

Length

The length of an input field should indicate how many characters the user can enter. The length of a field thus automatically corresponds to the maximum number of characters the user can enter.



The displayed length of especially long input fields can be reduced and the field can be made a scrollable field (see [Scrollable Input/Output Fields \[Seite 247\]](#)).

Character Set

For input data, the system standard is a non-proportional character set (equidistant script) which is required to indicate the length specification.

Output fields are usually displayed in proportional typeface.

Required Fields

Required fields are flagged by a question mark "?".

Defective Fields

Fields in which an error has occurred are highlighted by a different background color.

Fields With Possible Entries Pushbutton

Matchcode fields and other fields with possible entries have a [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#) to the right. The pushbutton may be permanent or not.

Appearance**Read-Only Fields**

If input fields cannot be changed in a particular dialog situation, the input field has the color of the window background. The three-dimensional display of the input field does not change in the process.

Read-only fields can be chosen with the active control indicator (Windows: `CTRL` arrow key) and the mouse. This allows the user to request help and to position the active control indicator in a selection table.

Design Guidelines

Input/output fields are usually assigned to a field name and positioned to the right of it, separated by at least two blanks, as long as they are not part of a table. Field names and I/O fields are arranged in blocks (see [Fields - Positioning \[Seite 253\]](#)) and can be collated in [Group Box \(Control Element\) \[Seite 280\]](#) with a group heading.

I/O Fields With Several Parts

You can assign several I/O fields to a field name if

- you can split an I/O field into several logical parts
- several related I/O fields are together on the screen template.

Try to split the field name in accordance with the I/O fields (see below). The individual parts of the input/output should be separated from one another by meaningful special characters. Place the special characters as close to the I/O fields as possible. Remember, however, that some special characters might have a different meaning in other countries.



Good:	<i>Plant/storage location</i>	_ / _		
Bad:		<i>Telephone</i>		(_) _

Explanation: The format for the telephone number shown above is common in Germany, but not in the U.S.A. In the United States, the area code and the phone number are separated by a hyphen.

Positioning Quantity and Currency Units

Place quantity/currency units to the right of the value. Leave only one blank between value and quantity/currency unit.

Exception: If task units are usually positioned to the left of the value, follow this arrangement to not confuse the user.

Check if you can include the unit in the field name (see example 2). Please keep the following points in mind:

- The quantity unit should appear in parenthesis.
- Insert a blank between field name and quantity unit.
- Make sure that there are at least two blanks at the end of the entire field name.



Good:	1)	<i>Material</i>	<i>Screws, nickel-plated</i>
		<i>Quantity</i>	20 kg
	2)	<i>Material</i>	<i>Screws, nickel-plated</i>
		<i>Quantity (kg)</i>	20

Explanation: If the quantity units were different in length, blanks would appear between the text and the parenthesis. In this case, it would make sense to assemble the quantity information

Design Guidelines

character by character within the program to achieve an consistent appearance with the second example above.

Displaying Output Values

Please note the following guidelines with regard to the output of values:

- Do not display leading zeros in numeric fields.



Good:	<i>Net amount</i>	234,78
	<i>Date</i>	03.02.90
Bad:	<i>Net amount</i>	000234,78
	<i>Date</i>	3.02.90

(The leading zero for the month is an exception to the above rule and should therefore be displayed.)

- For alphanumeric codes, split the character string into groups of three or four characters. This might cause problems if various SAP customers already have a different subdivision of the character strings.



Good:	<i>Code</i>	IDE7 938 IEH
Bad:	<i>Code</i>	IDE7938IEH

- Divide values with important subcomponent to emphasize the importance of these parts. Use special characters for the separation.



Good:	<i>Net value</i>	12.323.411,23
	<i>Date</i>	12.12.90
Bad:	<i>Net value</i>	12323411,23
	<i>Date</i>	121290

Number of the I/O Fields Per Field Name

- Every I/O field must have some sort of field name.
- If two fields are closely related, both can be assigned to the same field name. The field name does not have to refer to the subdivision.



Good:	<i>City</i>		—	—
--------------	-------------	--	---	---

(This example is only valid for certain countries; it does not apply to the U.S.A., for example.)

Design Guidelines

- If there are several I/O elements whose interrelation is not immediately clear, you should subdivide the field name accordingly. Do not use this approach for more than three fields.



Good:	<i>Plant/storage location</i>	_ / _
--------------	-------------------------------	-------

- You can also split up the field name into subnames. Follow this approach only if the subname does not have more than one or two characters and the subkey is properly structured. Use the colons as in the example.



Good:	<i>Terms of delivery</i>	1: _	2: _	3: _	4: _
--------------	--------------------------	------	------	------	------

Exception: The following fields have no field names:

- the field in which you can enter a set argument under a table
- selection columns

Short Descriptions

Place an explanatory text field for an I/O field next to the value, separated by one blank. Do not align the text field with other fields on the template if you have to move the text field away from the field to be explained.

Exception: If the vertical alignments on a template is interrupted more than three or four times, you should re-design the template.

"23451-4" is the value of the output field, "Screws, nickel-plated" is the short description:



Good:			
<i>Plant</i>	2	<i>Requisitioner</i>	Piper
<i>Material</i>	23451-4 <i>Screws, nickel-plated</i>		
<i>Text</i>	blah blah blah	<i>Value date</i>	X
Bad:			
<i>Plant</i>	2	<i>Requisitioner</i>	Piper
<i>Material</i>	23451-4	<i>Screws, nickel-plated</i>	
<i>Text</i>	blah blah blah	<i>Value date</i>	X

The explanatory text Screws, nickel-plated refers exclusively to the material number, that is, not to the other data in the right block. If you align the explanatory text, you give a wrong impression.

Explanation: Following the [Effects on Perception \[Seite 89\]](#), you are to give the user "visual aids" in the form of alignments. This guideline is, however, not as important as the unique assignment of the text field to the I/O field. In the bad example, the text field is also in a group to which it actually does not belong.

Design Guidelines

If there is not enough room for the explanatory text field to the right of the value, place the text field in a separate line. Define a separate field name for the text field.



Good:	<i>Material number</i>	23
	<i>Material name</i>	<i>Screws, nickel-plated</i>
	<i>Posting date</i>	03.26.1962
Bad:	<i>Material number</i>	23
		<i>Screws, nickel-plated</i>
	<i>Posting date</i>	03.26.1962

Positioning the Cursor

Depending on the task, the cursor should be automatically on the field the user generally uses most frequently.

Scrollable Input/Output Fields

Release R/3 3.0 introduces horizontal scrolling for input/output fields: The visible length of an input and/or output field can be smaller than the defined physical (ABAP/4 Dictionary) length. The user can scroll through the contents of the field using the mouse or the keyboard.

Visible and Defined Physical Length

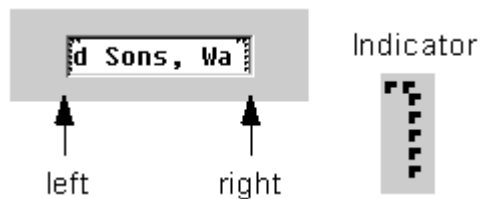
Scrollable input/output fields can be created by specifying a visible length value in the Screen Painter which is smaller than the defined physical length (see [Technical Notes \[Seite 249\]](#)).

Purpose

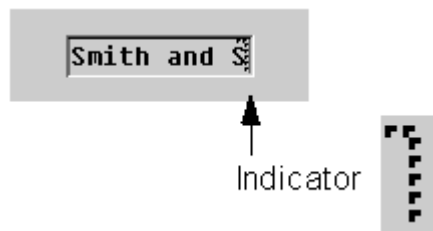
Scrollable input/output fields require less space on the screen because their visible length is smaller than the defined physical length. They are to be used when space on a window is limited. However, the scrollable field displays only a part of the full information and the user has to scroll to read the complete contents. For that reason, scrollable I/O fields should only be used when space reasons require them and they do not constitute much of a disadvantage for the user.

Appearance

The right border of scrollable I/O fields is interrupted, with input field also the left border.



Scrollable input field



Scrollable output field

Design Guidelines

In general, scrollable I/O fields should only be used if there is not enough space on a screen to display the full (physical) length of a field. As far as arrangement and appearance is concerned the guidelines listed under [Fields - Positioning \[Seite 253\]](#) apply.

Input Fields

- Typical key fields (for example, business area, material number)

Scrollable Input/Output Fields

The visible length can be smaller than the physical length to cover extreme field length requirements. As a rule, the visible length should, however, be long enough to display all important characters.

- (Currency) amounts and quantities (numeric fields)

If possible, the physical length should always correspond to the visible length because the user needs to see the complete figures to avoid errors and/or to recognize them immediately.

- Texts and descriptions

Input fields which allow the user to enter texts and descriptions can be displayed shorter than their physical length. You should, however, choose a length which requires only very little scrolling or which still allows to see the most significant characters.

Output Fields

- Explanatory text output field (check texts, displayed flat)

The visible length can be shorter than the physical length, but you should define a length that avoids unnecessary scrolling. The field should have a length which still allows the user to understand the meaning of the field.

Pump 200 KW, suction →

Check text

The arrow indicates that the check text is scrollable. Check texts are the foremost candidates for scrollable fields.



Check texts should always be available on a screen because they help the user to understand the meaning of the field. Scrollable output fields can even display check texts in situations when space usually does this not allow.

Do Not Use Scrollable Fields

Scrollable fields should **not** be used for date fields, quantity and currency specifications.

Technical Notes

Length of I/O Fields

The length of the I/O fields is defined in the ABAP/4 Dictionary.

“Scrollable” is one of the field attributes available in the Screen Painter which can be set for any field. If you set this attribute, the visible length visLg can be modified. The field is scrollable at runtime, when visLg is < defLg.

Matchcode Fields

The Basis Department is to provide automatic scrolling of a field when entering “=” (matchcode) in an input field, independent of the physically defined length because the matchcode can be longer than the normal field width. For the time being, the physical length of matchcode fields has to be defined wider than the actual visible length.

Overview of the Field Attributes

Overview of the Field Attributes

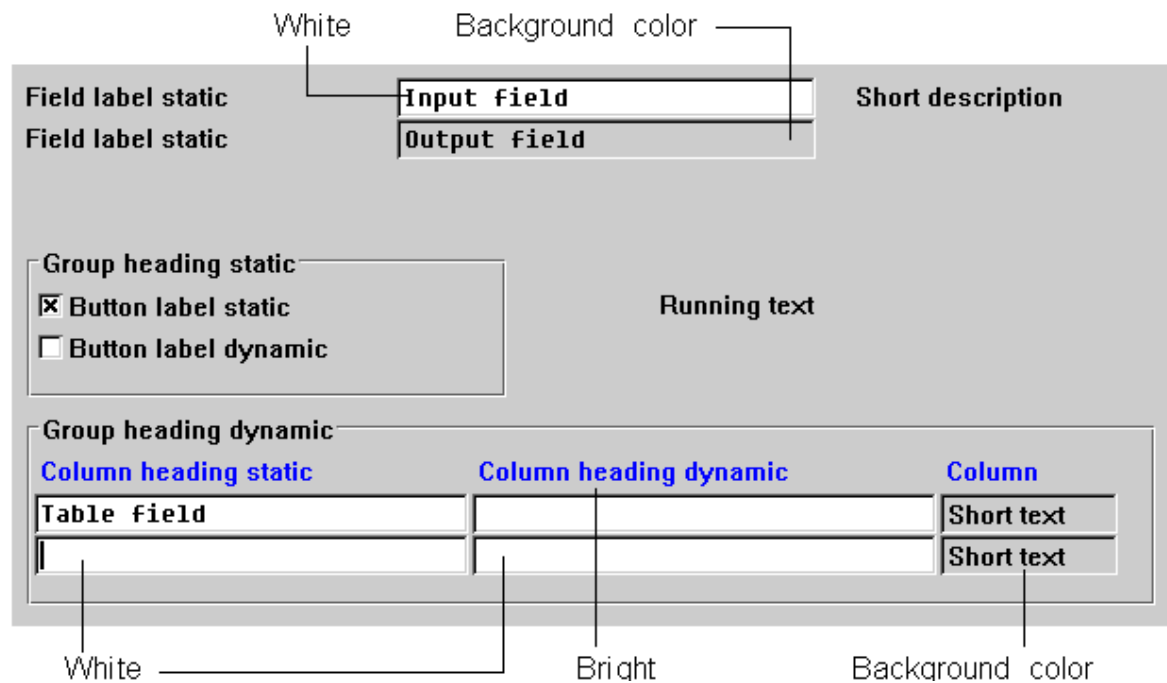
The following overview names the field type, the Screen Painter attributes, and shows exemplarily the appearance of the field type (see the following illustration) for correctly set flags. Important criteria are "flat vs. recessed" and "proportional script vs. equidistant script". Foreground processing is displayed in bold.

Interface object	Screen Painter flag to be set				
	E	A	FixA (1)	Equid (2)	Brigh
Input/output fields					
on screens	(+) (3)	(+)			
Output fields (4)					
on screens		(+)			
Names and texts					
Static field name					
Dynamic field name	(+)	+			
Static pushbutton name					
Dynamic pushbutton name		(+)	+		
Static group heading					
Dynamic group heading		(+)			
Continuous text (for example, dialog box)		(+)	+		
Check text (after input)		(+)	+		
Check text in step loops		(+)			
Static column title (loop)				+ (5)	+
Dynamic column title		(+)	+	+ (5)	+

Legend

1. In the Motif Screen Painter, FixA is called "output field only" (under the option "Program attributes").
2. Equid = equidistant script (not proportional typeface)
3. Bordered attributes have already been set; all others are to be set manually.
4. Output fields are fields which either the user or the system has filled with data and which can no longer be changed. This also includes units (for example, kg).
5. They are only to be set if the headings are a single field and cannot be split up into individual column headings.

Overview of the Field Attributes



Appearance of the individual field types with correctly set Screen Painter flags

Cross-References

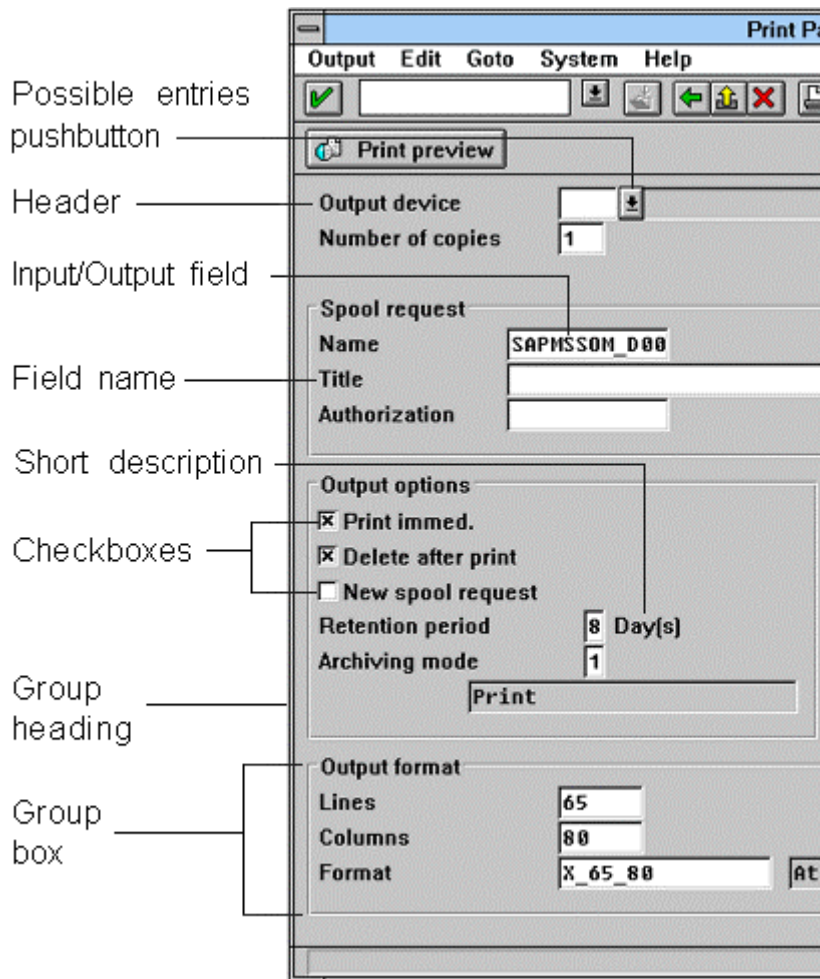
Cross-References

- [Effects on Perception \[Seite 89\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Field Names \(Control Element\) \[Seite 324\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)

Fields - Positioning

A good screen design depends largely on the arrangement of the fields within a work area. This section specifies field arrangement criteria for the current or adjacent - that is, the previous or next - screen.

Illustration



SAP GUI field design elements

[Arranging Fields According to Screen Sequence \[Seite 255\]](#)

[Grouping Several Fields in Blocks \[Seite 257\]](#)

[Gathering Several Blocks to Form a Block Group \[Seite 259\]](#)

[Arranging Several Block Groups on a Screen \[Seite 261\]](#)

[Cross-References \[Seite 266\]](#)

Arranging Fields According to Screen Sequence

Defining the Screen Sequence

The sequence of the screens should correspond to the user's work process. Place important fields on the earlier screens of a task.

Accessing Individual Screens of a Screen Sequence

Allow the user to address individual screens as variably as possible and to move backwards and forwards in a screen sequence.

Although this guideline affects the navigation between screens rather than the scroll design, it nevertheless affects the arrangement of information on screens.

Arrangement and Number of Fields on a Screen

A screen in a screen sequence should correspond to a concept or a user sub-task. Display only the data that are essential for a task in hand in the work area. Additional information should only be available on request or on a separate screen.

Arranging Fields in the Work Area

Field arrangement criteria:

- Follow the [Effects on Perception \[Seite 89\]](#).
- The sequence of screens should correspond to the user's work process.
- If there is a corresponding hard-copy form, the work area should have this form.
- Position general fields before specific fields.
- Position important fields in the top left corner of the work area.

If no special design elements such as different colors have been used in a work area, the attention of the user is distributed in accordance with the percentages shown in the following illustration. One-column work areas are thus preferable to two-column work areas.

40%	20%
25%	15%

Arranging Fields According to Screen Sequence

Distribution of Attention

- Generally, fields should appear at the top left of the work area. On initial screens, the first field should be implemented in the third line, on other data screens in the first line.

Keep Templates Consistent

Make sure that all colleagues involved in designing the screens for a particular task, all follow the same criteria. If a table has a header on one screen, the header is to appear on all screens in the task.

Efficiency vs. Good Design

In the event of conflict, an efficient dialog procedure has priority over an aesthetically pleasing design of the work area.

Grouping Several Fields in Blocks

Creating Blocks

Gather groups of field names and their respective I/O fields in blocks.



Both field names and I/O fields are left-aligned:

Posting date	<input type="text" value="06/29/1995"/>
Customer no.	<input type="text" value="123/9"/>
Address	<input type="text" value="Redwood City, CA"/>

Arranging Values to be Compared

Position fields whose values the user has to compare with each other one below the other.



From Customer no.	<input type="text" value="2345"/>
To Customer no.	<input type="text" value="341256"/>

Exception

If many "From-To" value ranges are grouped close together in the work area of a selection screen, the arrangement as shown in example 1 or example 2 is preferable. The blanks in the left field name of example 1 are tolerable in this case.



Good

Customer no.	From	<input type="text" value="123"/>	To	<input type="text" value="456"/>
Company code	From	<input type="text" value="0001"/>	To	<input type="text" value="0004"/>
Storage location	From	<input type="text" value="001"/>	To	<input type="text" value="299"/>

Customer no.	<input type="text" value="123"/>	To	<input type="text" value="456"/>
Company code	<input type="text" value="0001"/>	To	<input type="text" value="0004"/>
Storage location	<input type="text" value="0001"/>	To	<input type="text" value="299"/>

Bad

From Customer no.	<input type="text" value="2345"/>	To Customer no.	<input type="text" value="34567"/>
-------------------	-----------------------------------	-----------------	------------------------------------

Grouping Several Fields in Blocks

This guideline basically affects the output of values. With entry screens where numeric data is entered left-aligned, the different widths of the input values give some indication as to which is the largest and which the smallest value.

Repeating Name Elements in a Block

If two field names have one part in common, you should still use the full name for both field names.



Good

Material no.	123
Material group	0001
Posting date	06/29/1995

Bad

Material no.	123
group	0001
Posting date	06/29/1995

Aligning I/O Fields

The longest field name of a block determines where the I/O fields begin.



Good

Material no.	123
Material group	0001
Posting date	06/29/1995
Area	010

Bad

Material no.	123
Material group	0001
Date	06/29/1995
Area	010

Gathering Several Blocks to Form a Block Group

Number of Blocks, Lines and Fields per Block Group

Ideally, a block group contains only one block. If this is not possible, limit block groups to two or three blocks with six or seven lines at most and a total of ten to twelve input fields.

If there are a large number of fields, try to group them in two block groups.

Group Heading/Group Box of a Block Group

Identify a block group with a "group box" and a group heading.

You can also include dynamic fields in the heading if these are not too long (less than 15-20 characters).

Exception

If the contents of a block group is too heterogenous, you should omit the heading. Although headings such as "Other data" are perfectly acceptable in such circumstances.

Header data do not have a heading. They start in the first line of the work area (see illustration SAP GUI design elements for fields).



General data			
Material no.	678	23 Screws	
Posting date	06/29/1995		
Customer no.	123/9	Group	0001
Address	Redwood City, CA	Area	010
Document data			
Document/Reference	1212 / 45678	Value	x
Description	this is a text	Posting date	06/29/1995

Data for document no. 1213	
Document/Reference	1212 / 45678
Description	This is a text

In Example 2, a variable field is incorporated in the heading.

Using Separators In Block Groups

Do not use further separator lines within a block group. Do not use, for example,

Gathering Several Blocks to Form a Block Group

- vertical separators
- horizontal separators within a group box

Use blank lines to separate areas in block groups.

Minimum Horizontal Spacing Between Two Blocks in a Block Group

Leave at least five characters between two blocks.

Exception: If the longest field of the left block will most likely not be filled, you can reduce the space between blocks to one character.

Maximum Horizontal Spacing Between Two Blocks of a Block Group

Do not use a spacing of more than 10 to 15 characters between two blocks.

If you use [group boxes \[Seite 280\]](#), you can have blocks start at a fixed location, for example, at column 40. This can contribute to interface design consistency between applications.

Arranging Several Block Groups on a Screen

Blank Lines Between Block Groups

There is no blank line before a group heading.



General data			
Material no.	345	23 Screws	
Posting date	06/29/1995	Group	0001
Customer no.	123/9	Area	010
Document data			
Document/Reference	1212 / 45678	Value	x
Description	This is a text	Posting date	06/29/1995
Storage location			
Plant/Location	0001 / 0101		

Aligning Field Names in Different Block Groups

The second or third block in block groups does not have to be aligned with the corresponding second or third block in an adjacent block group. You should first arrange the blocks optimally within each block group (see above, Arrangement of several blocks to form a block group), and only then define the spacing for the blocks in the two adjacent block groups. If the distance is less than five to seven characters, you can align the leftmost block with the right-hand block. Otherwise, leave the left block as it is.



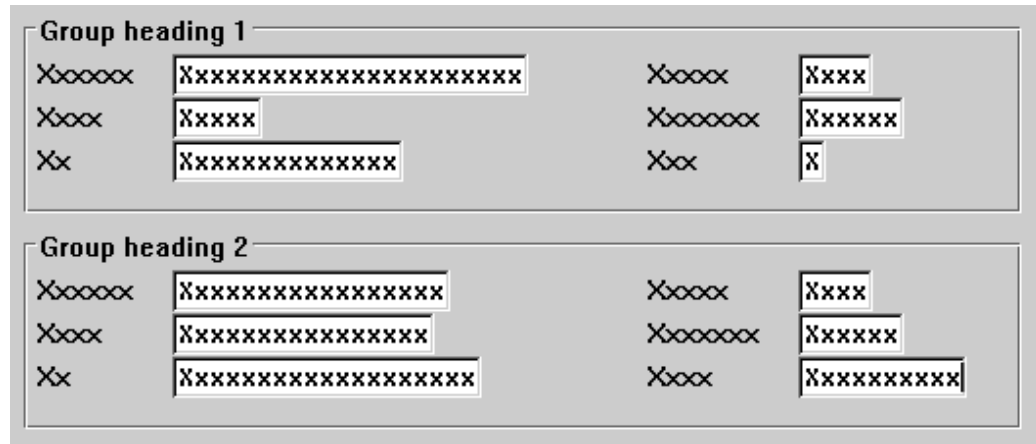
1. In schematic terms, the fully filled work area originally looks as follows:

Group heading 1			
Xxxxxx	XXXXXXXXXXXXXXXXXXXX	Xxxxx	Xxxx
Xxxx	XXXXX	Xxxxxx	XXXXXX
Xx	XXXXXXXXXXXX	Xxx	X
Group heading 2			
Xxxxxx	XXXXXXXXXXXXXXXXXXXX	Xxxxx	Xxxx
Xxxx	XXXXXXXXXXXXXXXXXXXX	Xxxxxx	XXXXXX
Xx	XXXXXXXXXXXX	Xxx	XXXXXXXXXX

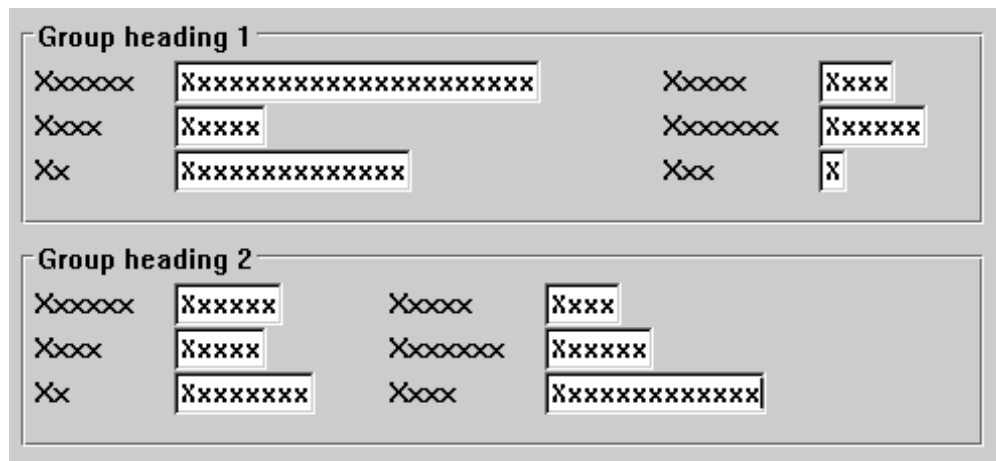
Arranging Several Block Groups on a Screen

In this case, you should align the two right-hand blocks with each other to achieve a more harmonious overall arrangement. Result:

Good and optimized:



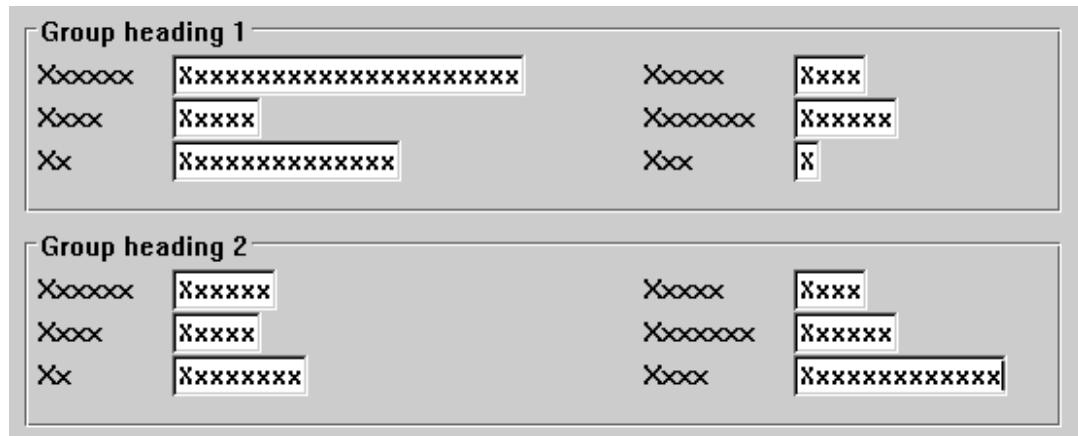
2. Schematically, the fully filled template looks as follows:



In this case, you should leave the lower right block close to the lower left block, rather than move it to the right to align it with the upper right block. As the following example shows, it is better to have the two groups close together:

Bad

Arranging Several Block Groups on a Screen



This guideline is a question of compromise between the spatial proximity of related data and alignments in the work area which give a more relaxed look to the entire screen.

Aligning Input/Output Fields in Different Block Groups

The above suggestion also applies to the positioning of input/output fields of various block groups.

Priorities

The alignment of I/O fields has priority over the alignment of field names.

This applies particularly to the leftmost blocks in the various block groups.

Do still try, however, to align both field names and input fields of the blocks in different block groups with each other.



Good

1. The left input fields in the three block groups are aligned vertically. The input fields of both right-hand blocks in the first and second block group are not vertically aligned with one another because the right-hand block of the first block group should be near to its left block.

Arranging Several Block Groups on a Screen

General data			
Material no.	678	23 Screws	
Posting date	06/29/1995	Group	0001
Customer no.	123/9	Area	010
Document data			
Document/Reference	1212 / 45678	Value	x
Description	This is a text		Posting date
06/29/1995			
Storage location			
Plant/Location	0001 / 001		

2. In contrast to the first example, the input fields of the right-hand blocks have also been aligned. The field names of the right blocks are not arranged one below another as one does not want to shorten the description in the upper block and the lower block cannot be shifted further to the left because of the long text field.

General data			
Material no.	678	23 Screws	
Posting date	06/29/1995	Customer group	0001
Customer no.	123/9	Customer area	101
Document data			
Document/Reference	1212 / 45678	Value	x
Description	This is a text		Date
06/29/1995			
Storage location			
Plant/Location	0001 / 001		

Bad

In this example, the groups of input field blocks and the field names are not vertically aligned.

Arranging Several Block Groups on a Screen

General data			
Material no.	345	23 Screws	
Posting date	06/29/1995	Group	0001
Customer no.	123/9	Area	010
Document data			
Document	1212 / 45678	Value	x
Text	This is a text		Posting date 06/29/1995
Storage location			
Plant/Location	0001 / 0101		

It is the input fields rather than the field names that should be aligned with one another, because the user (especially an experienced user) will tend to orient himself or herself "optically" towards the input values instead of the field names (which he or she already knows and therefore hardly reads consciously at all).

Maximum Number of Block Groups

You should have not more than three or four block groups on a screen.

Cross-References

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Effects on Perception \[Seite 89\]](#)
- [Field Names \(Control Element\) \[Seite 324\]](#)
- [Formulations \[Seite 601\]](#)
- [General Messages \[Seite 604\]](#)
- [Group Box \(Control Element\) \[Seite 280\]](#)
- [Hyphenation \[Seite 620\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Message Types and Error Groups \[Seite 611\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Uppercase/Lowercase \[Seite 619\]](#)

Function Keys - Overview

Function keys are special keys on the keyboard (usually referred to as "F<n>") which you can use to choose many functions in the R/3 System.

This section provides an overview of the function keys used in the R/3 System with their respective functions.

At first, the function key assignment as it was known until Release 3.0B is described. Then, the new extended function key concept to be introduced gradually from Release 3.0C on is described.

[Illustration and Purpose \[Seite 268\]](#)

[Overview - Current Conventions \[Seite 271\]](#)

[Guidelines - Current Conventions \[Seite 269\]](#)

[Guidelines - New Conventions \[Seite 273\]](#)

[Overview - New Conventions \[Seite 274\]](#)

[Technical Notes \[Seite 276\]](#)

[Cross-References \[Seite 277\]](#)

Illustration and Purpose

Illustration and Purpose

Illustration

Some keyboards have 24 special keys on them, however, many have only 12 or 8 special keys (F1– F24, F1–F12, or F1–F8. These keys are called **function keys** or **F keys**.

To choose a function on a keyboard with only 12 or 8 function keys, you press one or more modifier keys, such as CTRL or SHIFT, and then the function key.

With the new extended function key concept, you can assign functions to up to 99 function keys. The new function keys are activated by using the function keys used so far plus platform-specific key combinations (shortcut keys or menemonics), for example, CTRL+<character>.

Purpose

Function keys on the keyboard allow the user to access frequently used functions quickly. Function keys are an alternative to pushbuttons and menu options which are displayed on the screen and controled with the mouse (sometimes pushbuttons can also be operated with the keyboard).

Old: While the number of pushbuttons in the [Standard Toolbar \[Seite 307\]](#) and [Application Toolbar \(Control Element\) \[Seite 227\]](#) is limited by space, up to 24 function keys can be active.

New: While the number of pushbuttons in the Standard Toolbar and Application Toolbar (Control Element) is limited by space, up to 99 function keys can be active with the new function key concept.

Function Key Menu

The available function keys are listed in the [Function key menu \[Seite 278\]](#). To display the function key menu, click the right mouse button. You can choose a function from the menu.

Guidelines - Current Conventions

Groups

Function key groups were developed to standardize the function key settings and to make it as flexible as possible. They determine the way in which a function key in the corresponding group can be used.

Every function key belongs to one of the following four groups:

Group 1

SAP standard function keys: This group contains function keys which have the same function throughout the SAP System and which should almost **always be available**. If a function in this group is not active in a particular task, the corresponding function key must **never** be assigned to another function.

Group 2

Predefined function keys, if the function is active: This group contains function keys which should only be assigned to the functions specified for this group. Functions of this group do not need to be continuously active. If the function is not used in a task, the function key may be used for **another function**.

Group 3

Recommended function keys, if the function is active: This group contains function keys which **should** be assigned to the function specified here (recommendation), if active. The key can be used for another active function as long as the recommended function is not active in a task.

Group 4

Function keys with no specific function assigned to them: This group contains function which are seldom used.

Only Active With Function Key

Functions which were assigned function keys of groups 1 and 2 **cannot be active without the respective function key**. Functions of groups 3 and 4 can also be active without function key.

Functions Not Listed

Functions that are not listed (group 4) can be assigned to any available function key (except for those in group 1).

Functions Without Function Key

It should be noted that a number of functions are not to be assigned to **any** function key.

Consistent Function Keys

Within an application, try to use standard function keys for functions active in several of the application's tasks.

Overview - Current Conventions



For every function key, the function (or functions) assigned to it is specified. If several functions are possible, the sequence in which they are specified indicates the priority: If the first function is active, it is assigned to the function key. If it is inactive, then the second function (if active) is assigned to this function key, and so on.

The overview also lists the group to which a function key is assigned.

Function key	Function	Group
<F1>	Help	1
<F2>	Choose	2
<F3>	Back	1
<F4>	Possible entries	1
<F5>		4
<F6>		4
<F7>		4
<F8>		4
<F9>	Editor Long text Select/deselect	3
<F10>	Menu bar	1
<F11> and CTRL + S	Save Post	1
<F12>	Cancel	1
CTRL + P	Print	3
<F14>	Delete	3
<F15>	Exit	1
	End <task>	1
<F16>	Hold Save without check Sort	3
<F17>	<Object> to the left Other <object> Open	2 2
<F18>	<Object> up Previous <object>	2

Overview - Current Conventions

<F19>	<Object> down Next <object>	2
<F20>	<Object> to the right Other <object component>	3
CTRL + PAGEUP	First page	1
PAGEUP	Previous page	1
PAGEDOWN	Next page	1
CTRL + PAGEDOWN	Last page	1



[Overview of Standard Functions \[Seite 646\]](#) lists all R/3 standard functions in alphabetical order.

Guidelines - New Conventions

Virtual Function Keys

In the Menu Painter you can define up to 99 virtual function keys.

Physical Function Keys

The assignment of virtual function keys to physical function keys is platform-specific.

Display in Menus

For functions which are included in a menu and are assigned a function key, the corresponding key combination is displayed in the menu.

Function Key Menu

The function key menu lists the function keys v1 through v24.

Overview - New Conventions

Overview

Overview of the extended function keys

Virtual function key	Function
V1	Help
V2	Choose
V3	Back
V4	Possible entries
V10	Menu bar
V11	Save/Post
V12	Cancel
V14	Delete
V15	Exit
V52 through V69	(reserved for the ABAP/4 Development Workbench)
V70	Execute
V71	Find *
V72	Select all
V73	Deselect all
V74	Create
V75	Change
V76	Cut
V77	Copy
V78	Paste
V79	Undo
V80	First page
V81	Previous page
V82	Next page
V83	Last page
V84	Find next *
V85	Display
V86	Print *
V87 through V99	(reserved for future use)

*) Will be included in the standard toolbar from Release 3.0C on.

Reserved Function Keys

Function keys V52 through V69 are reserved for the ABAP/4 Development Workbench, function keys V87 through V99 for future (system) extensions.

Function Keys No Longer Predefined

The *Select* function is no longer predefined. If required, please use function key V9.

New Virtual Function Keys

If new virtual function keys are to be added, use function keys V25 through V48 as well as V21 through -V24.

Technical Notes

Conversion



The old function key assignment still applies in Release 3.0. The new assignments are available **in addition** to the existing assignment. From Release 4.0, the new extended function key concept will replace the existing concept.

Platform-specific Characteristics

For platform-specific characteristics of the new function key concept, please refer to the separate documentation.

Function Key Display

The display of the function keys and/or key combinations is language- and platform-specific:

Display in the function key menu

Key	German	English	Apple
Ctrl	STRG	CTRL	<Ctrl>
Shift	UMSCH	SHIFT	<Shft>
Alt	ALT	ALT	<Alt>

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Defining Functions \[Seite 456\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function key menu \[Seite 278\]](#)
- [Overview of Standard Functions \[Seite 646\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Function Key Menu

Function Key Menu

The function key menu can be opened by clicking the right mouse button. It displays all active function keys and allows to choose a function from it.

Illustration

Help	F1
Choose	F2
Back	F3
Possible entries	F4
Selection criteria..	F5
Change user master	F6
Totals variant	F7
Delete sel.criteria	F8
Select/deselect	F9
Exit	F15
Line layout...	F16

The function key menu; the *Exit* function is chosen

Purpose

The function key menu is implemented as a pop-up menu. It shows a list of all available function keys and allows to choose a function from it.

In the primary window, it contains all defined function keys, including the keys already displayed in the [Standard Toolbar \[Seite 307\]](#) and [Application Toolbar \(Control Element\) \[Seite 227\]](#). The pushbuttons defined in the Screen Painter are also displayed in the dialog box.

Appearance

Arranging Options in the Function Key Menu

The function names and function keys are defined as a list in the function key menu as follows:

- "Function" left-aligned, "Function key" right-aligned, for example:
"Detail screen F7"

Sequence of Options

- ascending ordinally
- After F4, F8, F12, F16 and F20, there is a separator line, if keys in the respective group are available.



With the introduction of the new extended function key concept, the function key menu displays the virtual function keys v1 through v24 in ascending order.

Use/Functionality

If the right mouse button is pressed, a pop-up menu is displayed at the mouse pointer position. This also applies to modal dialog boxes.

To activate a function from the function key setting, position the mouse pointer on the desired function and click the left mouse button. Users who prefer to use the keyboard use the arrow keys to move the focus to the required function key and then press ENTER to initiate the function.

Technical Notes

The functionality of the function key menu is made available by the Basis Department.

For more information on the design of function key and pushbutton texts, please refer to [Application Toolbar \(Control Element\) \[Seite 227\]](#).

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Control Elements - Overview \[Seite 168\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Modal Dialog Box \[Seite 65\]](#)

Group Box (Control Element)

Group Box (Control Element)

Group boxes collate related groups of fields or control elements. They are an important screen design element.

Illustration

The illustration shows a SAP screen layout with the following components and labels:

- Header (no group box):** Material ICEBOX Icebox, 5 gall
- Group heading:** Basic data
- Group box:**
 - Base unit of measure PC Piece
 - Old material number
 - Division 98
- Group heading:** General data
- Group box:**
 - Basic material
 - Prod./insp. memo
 - Net weight 2.5
 - Gross weight
 - Volume
 - Size/dimension
 - EAN/UPC

Group boxes on a screen

[Purpose and Appearance \[Seite 281\]](#)

[Design Guidelines: Basic Rules \[Seite 282\]](#)

[Design Guidelines: Arrangement \[Seite 283\]](#)

[Design Guidelines: Screen Examples \[Seite 285\]](#)

[Technical Notes \[Seite 288\]](#)

[Cross-References \[Seite 289\]](#)

Purpose and Appearance

Purpose

Group boxes are a means for achieving a clearly structured screen design. They collate related information together in a box and identify it with a group heading.

Appearance

The platform-specific design is used for group boxes.

Headings

Every group box should have a group heading. Do not highlight the headings. Headings can also be set at runtime.

Design Guidelines: Basic Rules

Design Guidelines: Basic Rules

Larger groups of information units which are to be separated from one another are placed in a group box.

Group boxes only contain particular groups, not the entire screen.

Exceptions

Some information blocks do not require a group box.

- **Small Information Blocks**

Small information blocks which do not necessarily require separation from other information on the screen can be separated with a blank line.

- **Header Data**

Do not use group boxes for header data. Use a blank line to separate header data from other information on the screen.

- **Dialog Boxes**

Dialog boxes with only one information group do not contain a group box.

Subscreens

When using subscreens, the group box should be defined in the calling screen because two extra lines are to be specified for the box in the subscreen definition and the size of the area in the calling screen would have to be increased correspondingly.

Hiding

Group boxes can be hidden at runtime if all fields contained are also hidden.

Design Guidelines: Arrangement

Alignment

Define group boxes in accordance with the following guidelines. The guidelines are listed in terms of priority:

1. The screen can often be divided into two large columns. Place the right border of the left-column box at the center of the screen (as far as column 40). Place the right border of the right-column boxes which extend almost to the edge of the window (from about column 60, that is) at the right window border.
2. Adjacent boxes should be aligned, if this can be achieved by extending the border by a few blank columns or lines (orientation value: 2-3 blank lines or 5-10 blank columns).
3. If the first two guidelines are not applicable, boxes should be placed directly around the fields of the group.



With fields without a permanently displayed [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#), make sure to leave spacing of at least three columns after the field so that the group box is not obscured by the possible entries pushbutton.

Arranging Groups and Individual Elements

Horizontal Arrangement

If two groups appear next to each other, they should be regularly distributed on the screen. To do so, place the border of the box at the screen center and/or the screen border. The boxes should be separated by approx. 1-2 columns. Larger boxes should be positioned to the left.

Text without group box (particularly header data) starts in column 1; this also applies to dialog boxes. The space between window border and first column is set centrally by SAP GUI. Texts and selection elements ([Checkboxes \(Control Element\) \[Seite 176\]](#), [Radio Buttons \(Control Element\) \[Seite 193\]](#)) that are part of the box start in the column directly following the box.

[Input/Output Fields \(Control Elements\) \[Seite 238\]](#) should be left-aligned. This also applies to the header data and the following groups.

Vertical Arrangement

Two group boxes positioned below each other are not separated by blank lines. Initial screens are an exception, because a blank line can be inserted between the boxes to achieve a regular distribution.

Separate the header data of the following group always by a blank line.

Last Check...

When you have defined the group box conforming to the above mentioned guidelines, check the screen if some of the elements need to be corrected slightly for an visually pleasing result. A good overall impression has priority.

Design Guidelines: Screen Examples

The following illustrations collate the design guidelines for initial screens, typical data screens, and tables.

Header data:
- start in row 3
- no group box

Object to be copied:
- group box

Selectable object components as checkboxes
(multiple selection)
or as radio buttons
(single selection)

The screenshot shows an SAP initial screen with the following elements:

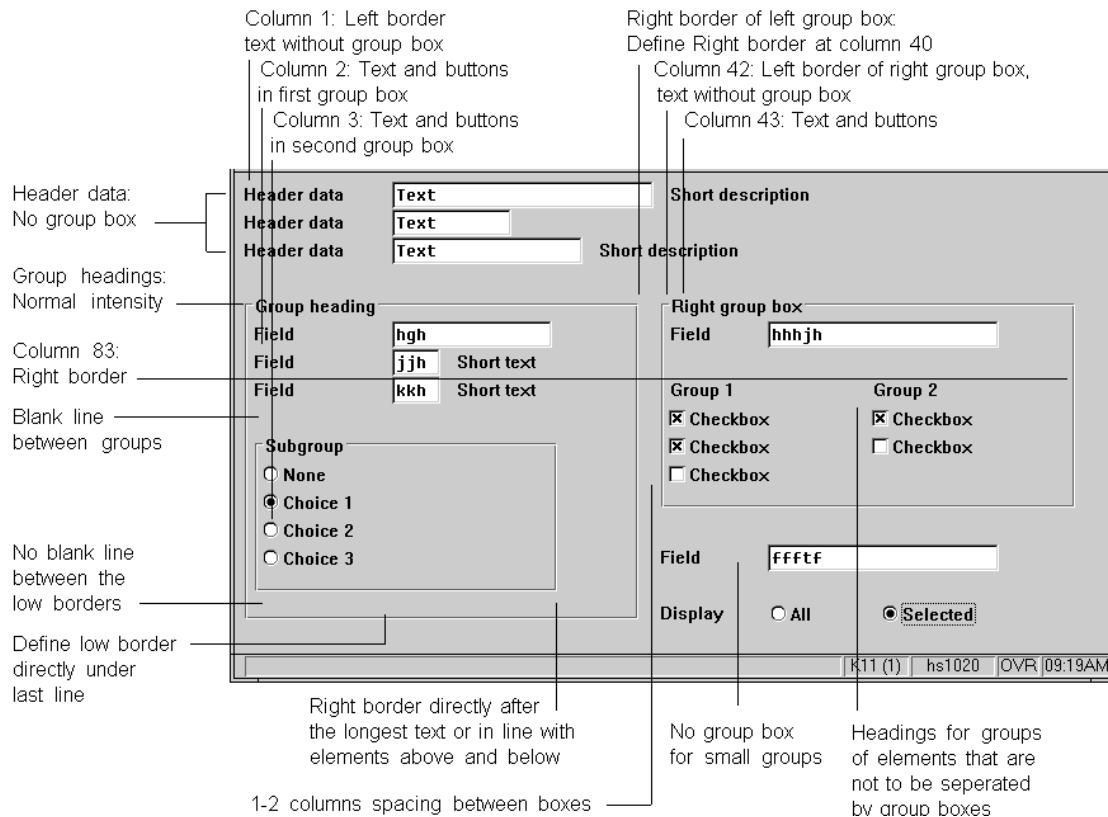
- Header data:** A group box containing three input fields: "Material" with value "Screw-47/12", "Plant" with value "0001", and "Storage loc." which is empty.
- Object to be copied:** A group box containing one input field: "Material" with value "Screw-47/01".
- Selectable object components:** A group box titled "Views" containing five checkboxes: "Classification" (checked), "Purchasing" (unchecked), "Engineering/Design" (unchecked), "Forecasting" (checked), "Storage" (unchecked), and "MRP" (unchecked).

Annotations with lines pointing to the screen elements:

- Header data: - start in row 3, - no group box (points to the header data group box).
- Object to be copied: - group box (points to the "Copy from" group box).
- Selectable object components as checkboxes (multiple selection) or as radio buttons (single selection) (points to the "Views" group box).

Designing an initial screen

Design Guidelines: Screen Examples



Designing a data screen

Column headings highlighted

Checkboxes for selecting rows

Table displayed in group box

Optional:
Page display

Table heading			
	Column 1	Column 2	Column 3
<input checked="" type="checkbox"/>	cddfc d	ddfd	fffd
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input checked="" type="checkbox"/>	tzui	uuui	uuui
<input type="checkbox"/>			
<input type="checkbox"/>			

Entry 1 of 27

Designing screens using entry tables (step loops)

Technical Notes

Technical Notes

Group boxes and group headings are made available by the Basis Department.

[Input/Output Fields \(Control Elements\) \[Seite 238\]](#) provides an overview of the Screen Painter flags.

Cross-References

- [Control Elements - Overview \[Seite 168\]](#)
- [Effects on Perception \[Seite 89\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)

Long Screens

Long Screens

In Release 2.1, the standard screens size was 21 lines and 83 columns. As of Release 3.0, the maximum number of lines that can be defined for a screen is 200 lines, the maximum number of columns is 240. In addition, hidden fields are removed at runtime which means that the number of visible lines is smaller than the number of lines defined in the standard.

This describes the screen size definition in the light of the new 3.0 functionality. In the following, screens that exceed the standard screen size are referred to as 'long screens'. To support the handling of long screens, the Basis Department will provide improved scroll bar and micronavigation functionality.

[Use \[Seite 291\]](#)

[Screen Design \[Seite 292\]](#)

[Checks \[Seite 293\]](#)

[Cross-References \[Seite 294\]](#)

Use

The use of long screens depends on the following three aspects:

Task Structure

- Long screens should contain only data of the same hierarchy level, for example, overview data.
- Hierarchically structured tasks with detail screens should use separate screens for dependent detail data display. (Remark: As soon as the technical possibilities are available, these detail screens should be defined as modeless dialog boxes). The detail data, in turn, can be arranged on long screens.

Logical View

The most important criterion for grouping fields on one or more screens should be the number of fields the customer needs to access quickly at the same time per business process (within the task). The set of fields in question is referred to as “logical view”.

- All important data of a logical view is to be positioned on one screen.
- Less important or rarely used data of a logical view is to be transferred to a dialog box.
- The compression effect has low priority in favor of a meaningful arrangement of the data and an efficient navigation to these screens.
- This means especially that all data of different views of a task should be positioned on a long screen, not only for compression reasons.

Editing Mode

- Generally, the user should be able to access the desired field quickly. This requirement results in the following screen size recommendations:
- For the sequential creation of master data, long screens make sense as long as meaningful data views per screen can be achieved.
- To change and/or flexibly edit data on long screens, you should allow the user to navigate directly to smaller units. This can be achieved, for example, by selecting the desired group on the initial screen and positioning the cursor in the corresponding group (Positioning the cursor via the frontend in the visible area, which goes along with it, still has to be realized).
- If editing data is the primary edit mode (because data has been entered automatically, for example), you should design small meaningful and flexibly accessible screens.

Screen Design

Screen Design

Number of Lines/Columns

- You should not design views/screens with more than 50 to 60 visible lines because orientation might suffer. Depending on the average number of hidden fields the customer uses, the screen can actually have more than 50 to 60 fields at the time of designing the screen.
- Required horizontal scrolling in combination with vertical scrolling on a screen complicates the handling and should be avoided. You should not design screens which exceed the standard number of lines and the standard width of 83 columns.

Compression

- All screens should be set to “Compression on” by default.
- Release 3.0 should be delivered with active compression algorithm, even for standard sized screens.

Field Arrangement

- Related fields should be arranged in group boxes, with groups not larger than the maximum visible part of the screen.
- To achieve optimum compression, a two-line layout should only be used for fields which are usually not hidden or only together with other fields.
- With multiple-line fields, you should position fields which are presumably hidden frequently at the right screen side to avoid “gaps” on the left border.
- Any dynamic extensions of long screens generated automatically by the system depending on previous data input do not make sense because they do not always appeared at the same position on the screen. Additional data should be displayed in modeless dialog boxes, by reading the relevant parameters before the screen is sent or by a user-driven screen extension after a pushbutton has been activated.

Loops

- On long screens, you are to define only fixed loops and table controls.
- You should choose a loop size which allows to display the entire loop on a screen. (If you exceed 21 lines, the Screen Painter will issue a warning.)
- If you do not use any table controls, you can optionally provide each loop with a separate scroll bar in the work area.

Checks

- The entire screen is checked when the user activates `ENTER` or when OK Code events occur.
- The Basis Department will provide the functionality to check by group boxes in a subsequent 3.0 Release. Checking by group boxes can then be supported by the task developer, if checks have been performed in the corresponding group box or check texts have been displayed for fields (for example, explanatory texts on the template).

Cross-References

Cross-References

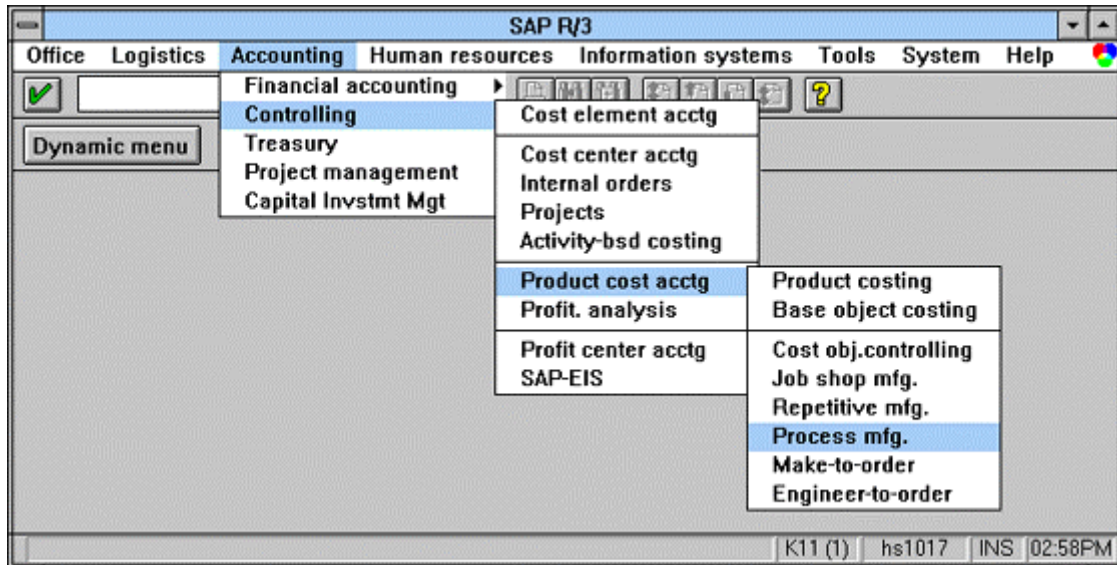
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Fields - Positioning \[Seite 253\]](#)

Menus (Control Elements)

The R/3 System is a menu-driven system which uses two types of menus

- the menu bar
- pull-down menus

Illustration



Menu bar with open pull-down and cascading menus

[Purpose \[Seite 296\]](#)

[Menus in the R/3 System \[Seite 297\]](#)

[Shortcuts \[Seite 298\]](#)

[Cross-References \[Seite 299\]](#)

Purpose

Purpose

[Menüs - allgemeine Gestaltung \[Seite 548\]](#)

Advantages of Menus

Menus are designed to provide an efficient method for accessing functions: The user does not have to remember and enter commands, because all the available functions can be selected from menus. The user only has to remember what the functions offered in the menus do.

Hierarchical Order in Menus

If the number of functions is very large, it is necessary to group the functions hierarchically. The functions are, for example, grouped together under generic terms listed in the menu bar. This can be implemented on several layers as in the R/3 System (see below.).

Finally, the hierarchy can be extended by means of cascading menus.

Overview gives design guidelines for the correct setup of menus.

Disadvantages of Menus

A hierarchical structure of functions has the disadvantage that the user must travel long ways through the system to access a particular function. Moving through the menu hierarchy can be complicated and time-consuming. Methods to overcome this shortcoming are described below.

Menus in the R/3 System

The menu structure in the R/3 System is based on an object-oriented approach (see [Object-Oriented User Interface \[Seite 129\]](#)). This approach helps the user to navigate efficiently in the extensive R/3 System and allows a quick access to the functionality.

The R/3 System uses two types of menus,

- the **menu bar**
- pull-down menus

Both menu types play an important role in the R/3 System: The menu bars provide the structure, the pull-down menus offer the functionality.

Menu Bars

The R/3 menu system is made up of three categories of menu bars (see [Levels in the SAP System \[Seite 72\]](#)):

- the SAP standard initial menu on the main menu level for choosing the application
- many menus on object class level or application level that contain the different objects of the application
- a menu for each task that is the same for all applications.

Please refer to the keywords listed below for information on the design of the menus.

Menus of the Task level

The largely standardized structure of the menus at task level is an essential utility for accessing the functionality of the R/3 System. Consistency on this level guarantees a standard behavior within and across tasks and applications. This reduces the memory load of the user considerably.

Shortcuts

Shortcuts

Despite the specific menu structure used in the R/3 System, the user has to travel through several menus before the desired functionality can be accessed (see [Accessing Tasks \[Seite 77\]](#)). This is intolerable for experienced users or users who only require a small portion of the overall functionality. Therefore, there are a number of "shortcuts" which allow the direct access to the required functionality in the R/3 System:

- **User defaults:** The user can define the start menu which is displayed right after logging on.
- **Access characters:** Particularly, experienced keyboard users prefer to enter mnemonic access characters to quickly access the required functions.
- **Fast path:** The sequence of access characters which leads directly to the desired task can be entered into the command field following a period.
- **Transaction codes:** Tasks can be called directly by entering transaction codes in the command field. Disadvantage: The user must know and remember the specific transaction code of a task (the command list proves to be a useful tool in this respect) (see [Command Field \[Seite 201\]](#)).
- **User menu:** In the user menu, a window which is active in parallel to the R/3 primary window, a user can define frequently used applications and tasks. Once defined, the user can choose the required tasks directly from the user menu. This procedure is particularly worthwhile when a user is active in many applications and tasks.

Cross-References

- [Access Characters \[Seite 184\]](#)
- [Accessing Tasks \[Seite 77\]](#)
- [<Application> \(Menu\) \[Seite 564\]](#)
- [Command Field \[Seite 201\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Environment \(Menu\) \[Seite 583\]](#)
- [Extras \(Menu\) \[Seite 585\]](#)
- [Goto \(Menu\) \[Seite 580\]](#)
- [Levels in the SAP System \[Seite 72\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Object-Oriented User Interface \[Seite 129\]](#)

Check Text (Control Element)

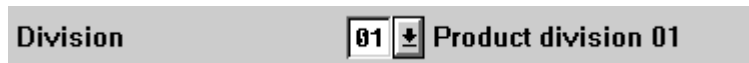
Check Text (Control Element)

Check texts or explanatory texts are display-only fields. In general, they follow an input field and explain the input data in more detail, for example, the meaning of a numeric or alphanumeric key.

Guidelines

Appearance

Check texts appear flat on the template and have the same color as the background. If the user has entered check texts, check texts can also appear recessed and in the color of the background, for example on follow-up screens.



Checktext for an entry field (right)

Check texts can be scrollable.



Scrollable check text

Arrangement

In general, check texts appear next to an input field (separated by one blank character, when there is no Possible entries pushbutton).

If a non-permanently displayed Possible entries pushbutton is displayed, leave three blank characters between input/output field and the check text to the right of the field.



The Screen Painter makes it impossible to use the first three characters after a field.

Alignment

Check texts (explanatory texts) should not be aligned if this positions them too far away from the corresponding fields. In general, leave one blank character between the Possible entries pushbutton and/or the input field without a possible entries pushbutton and the check text.

Scrollable Check Texts

Check texts are the foremost candidates for scrollable fields. An arrow indicates that the check text is scrollable (s. [Scrollable Input/Output Fields \[Seite 247\]](#)).

Cross-References

- [Field Names \(Control Element\) \[Seite 324\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#)

Status Display (Control Element)

Status Display (Control Element)

Status displays are icons which indicate the state of an object or a value.

Illustration



Status display

Use

You can use icons to indicate a status (processing state of an object). For example, you can implement the status display of a table entry (checked, erroneous) by using a two-value display. Another example is the in-box in the mail system: if a message has been viewed, a checkmark could appear next to the message (implicit two-value display). More complex, three-value displays are also possible.


With digital status displays we distinguish the following cases:

- Value (implicit 2-value, 3-value, etc displays)
- Generic meaning (for example, On - Off) vs. specialized meaning (for example, volume, high or low)

Appearance







With generic meanings we distinguish the following status displays:

- 2-value display
 - One** pictorial representation: checkmark. Meaning: for example, "okay", "done" or "viewed".
 - Two** pictorial representations: checkmark and cross. Meaning: "yes" and "no", "done" and "to do", "okay" and "not okay" etc.
 - Two** pictorial representations: Plus and minus signs. Meaning: "positive" and "negative"
 - Two** pictorial representations: Lock open or closed. Meaning: "locked" und "not locked"
- 3-value display
 - Pictorial representation: Traffic lights red - yellow - green. Meaning: "negative", "neutral" and "positive"
 - Pictorial representation: Checkmark - cross - lightning. Meaning: "okay", "open", "failure"







	CHECKED	Checked			
---	---------	---------	--	--	--

Icons as status display: implicitly two-state

Status Display (Control Element)

	CHECKED	Yes		INCOMPLETE	No
	LOCKED	Locked		UNLOCKED	Unlocked
	POSITIVE	Positive		NEGATIVE	Negative

Icons as status display: two-state (icons in rows belong together)

	CHECKED	Checked		GREEN_LIGHT	Positive
	INCOMPLETE	Incomplete		YELLOW_LIGHT	Neutral
	FAILURE	Failure		RED_LIGHT	Negative

Icons as status display: three-state (icons in columns belong together)

Icons for status displays

Design

Place status displays always close to the corresponding control elements.

With field names, checkboxes etc. the status displays is always positioned to the right of the corresponding control element.

Arrangement in Tables

- at the beginning of a row, if the status display refers to the entire row
- in any column, if the status display does not refer to a specific object



In general, you have to define a context-specific and translatable quickinfo text.

Technical Notes

Status icons are defined as templates, that is, they should have a height of one line and a width of two characters (traffic lights four characters). In the graphical Screen Painter there is a special object for status displays in the toolbar.

Cross-References

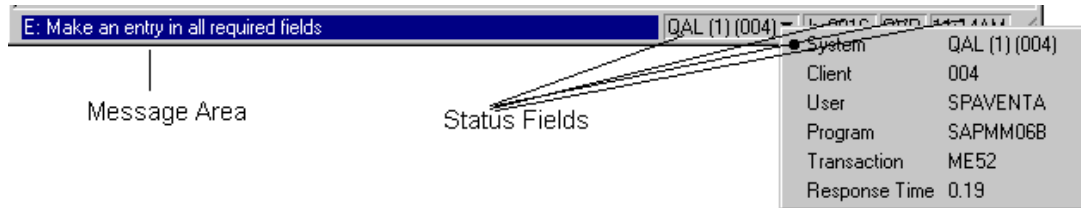
- [Control Elements - Overview \[Seite 168\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)

Status Display (Control Element)

Status Bar

The bottom line of the R/3 primary window contains the status bar that displays messages or information on the status of the application.

Illustration



The status bar

Purpose and Appearance

Purpose

The system displays messages and status information in the status bar.

Position

The status bar is located at the bottom edge of the R/3 primary window.

Structure

The status bar contains

- messages at the left of the status boxes; information on a chosen function (of the [Standard Toolbar \[Seite 307\]](#) or the [Application Toolbar \(Control Element\) \[Seite 227\]](#)) are also displayed here.



Depending on the dialog situation and error type, error messages can also be displayed in a separate dialog box.

- status information to the right of the message:
- The first status box contains a popup window from which you can select to display the following information:
 - System
 - Client
 - User
 - Program
 - Transaction
 - Response time
- The second status box displays the server to which you are connected.

Status Bar

- The third status box specifies your data entry mode. By clicking on the box with the mouse, you can toggle between overwrite and insert modes.
- The fourth status box displays the time. By clicking on the box with the mouse, you can toggle between the clock time and the system response time for each transaction.

Technical Note

The functionality of the status bar is provided by the Basis Department.

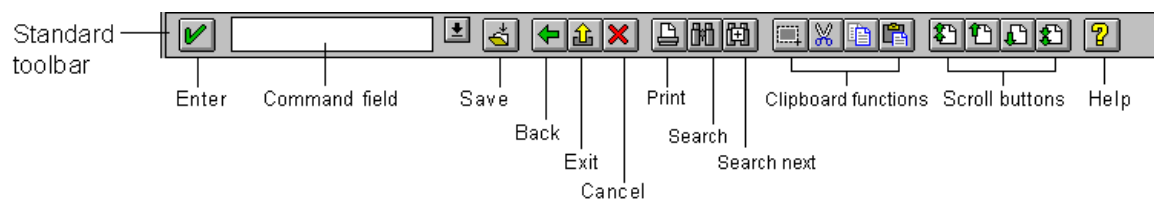
Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Control Elements - Overview \[Seite 168\]](#)
- [General Messages \[Seite 604\]](#)
- [Message Types and Error Groups \[Seite 611\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Standard Toolbar

The standard toolbar in the R/3 primary window displays pushbuttons with symbols representing general functions. The user can quickly choose the functions offered in the standard toolbar by clicking on them. The standard toolbar also contains the command field.

Illustration



The standard toolbar (3.0d implementation with clipboard functions)

[Purpose and Appearance \[Seite 308\]](#)

[Design Guidelines \[Seite 310\]](#)

[Icons for the Standard Toolbar \[Seite 311\]](#)

[Technical Notes \[Seite 312\]](#)

[Cross-References \[Seite 313\]](#)

Purpose and Appearance

Purpose and Appearance

[Befehlsfeld \[Seite 201\]](#)

[Auswahlbuchstaben \[Seite 184\]](#)

[Erste Seite \[Seite 505\]](#)

Purpose

The standard toolbar contains pushbuttons for important cross-application functions that can be quickly accessed with the mouse.

It also contains the command field for entering a fast path (see Pushbuttons with Icons on Initial Screens (Tools)) and transaction codes.

Standard Toolbar vs. Application Toolbar

The standard toolbar includes general (generic) functions of a cross-application nature. Application-specific functions are in the [Application Toolbar \(Control Element\) \[Seite 227\]](#).

Appearance

The standard toolbar is only displayed in primary windows. It is displayed directly under the menu bar and can never be hidden.

The standard toolbar contain pushbuttons with a graphical label, while the application toolbar contains pushbuttons with textual labels.

Structure

The standard toolbar contains the following graphical pushbuttons with symbols and control elements in the order specified:

1. OK/Check = ENTER (see [Continue \(Function\) \[Seite 540\]](#))
2. the [command field \[Seite 201\]](#) implemented as drop-down list which provides a history of the last 20 commands entered by the user; the drop-down list can be displayed by clicking on the respective pushbutton;
3. the [Save or Post \(Function\) \[Seite 524\]](#) pushbutton,
4. the pushbuttons for the three navigation functions [Back \(Function\) \[Seite 544\]](#), [Exit or End <application> \(Function\) \[Seite 492\]](#), and [Cancel \[Seite 470\]](#),
5. the pushbuttons for the group *Search and Print*: [Print \[Seite 496\]](#), *Search*, and *Search next*,
6. the pushbuttons for the local clipboard functions [Cut \[Seite 486\]](#), [Copy \[Seite 513\]](#) and [Paste \[Seite 502\]](#); for Windows and Apple, the pushbutton for *Select mode* is positioned at the beginning of this group
7. the four scroll buttons [First Page, Last Page \(Function\) \[Seite 505\]](#), [Previous Page \[Seite 535\]](#), and [Next Page \[Seite 535\]](#),
8. for Windows 95 and Windows NT the pushbutton for accessing the settings of the *Session Manager*,
9. and, finally, the F1=Help pushbutton (see [Help \(Function\) \[Seite 509\]](#)).

All pushbuttons of a group are close together.

Design Guidelines

Design Guidelines

Naming Pushbuttons

The texts of the pushbuttons in the standard toolbar consist of the function name and the term "pushbutton", for instance:

- *Enter* pushbutton
- *Save* pushbutton
- *Help* pushbutton

Symbols

Every symbol has a predefined position; if a function is inactive, the corresponding pushbutton is dimmed.

Using the Keyboard

To operate the system with the keyboard, all pushbutton functions are also assigned to function keys.

Standard Toolbar Help

If the user clicks on a symbol and keeps the mouse button pressed, the system displays a description of the function and the corresponding F-key in the status bar, for example, "Exit F15".

Clipboard Functions, Selection Mode

The standard toolbar contains the local clipboard functions *Cut*, *Copy*, and *Paste*. The central functions [Cut \[Seite 486\]](#), [Copy \[Seite 513\]](#) and [Paste \[Seite 502\]](#) are located in the [Application Toolbar \(Control Element\) \[Seite 227\]](#).

The *Selection mode* pushbutton is only available under Windows and Apple.

Layout Menu

The icon for the layout menu is the last entry in the menu bar, on the right. The functions of this menu allow the user to modify the appearance of the R/3 window.

Pushbuttons No Longer Available

The F4=[Possible Entries \(Function\) \[Seite 499\]](#) button is no longer in the standard toolbar. The functionality of the Possible entries pushbutton has made the inclusion of this button superfluous.

Icons for the Standard Toolbar



















More icons for system-wide functions have been added to the standard toolbar. You can also use icons in the application toolbar of primary windows and in dialog boxes.

Standard Functions in the Standard Toolbar

If specific functions are defined as function keys in the Menu Painter, they are automatically included as icons in the standard toolbar (not for Release 3.0A):

Print (F13); *Scroll* (*First Page*, *Previous Page*, *Next Page*, *Last Page* - F21 through F24)

Other functions such as *Mark* (Windows and Apple only), *Copy*, *Cut* and *Paste* are also included in the standard toolbar and assigned local functionality. These also include the system function *Create session*.

	SAVE	<i>Save</i>		EXIT	<i>Exit</i>
	BACK	<i>Back</i>		CANCEL	<i>Cancel</i>
	FIRST_PAGE	<i>First Page</i>		PREVIOUS_PAGE	<i>Previous page</i>
	LAST_PAGE	<i>Last Page</i>		NEXT_PAGE	<i>Next page</i>
	SYSTEM_LOCAL_CUT	<i>Cut (local)</i>		SYSTEM_LOCAL_COPY	<i>Copy (local)</i>
	SYSTEM_LOCAL_MARK	<i>Select mode (local)</i>		SYSTEM_LOCAL_PASTE	<i>Paste (local)</i>
	SEARCH	<i>Find</i>		SEARCH_NEXT	<i>Find next</i>
	SYSTEM_PRINT	<i>Print</i>		SYSTEM_USERMENU	<i>Session Manager</i>
	SYSTEM_SETTINGS	<i>Layout menu</i>		HELP	<i>Help</i>

Icons for standard toolbar functions

Technical Notes

Technical Notes

The functionality of the standard toolbar is made available by the Basis Department.

Functions that are displayed in the standard toolbar are normally defined in the Menu Painter. The system determines automatically whether and in what position a function is to be displayed in the standard toolbar.

For information on defining pushbutton texts, please refer to [Application Toolbar \(Control Element\) \[Seite 227\]](#).

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Back \(Function\) \[Seite 544\]](#)
- [Cancel \(Function\) \[Seite 470\]](#)
- [Command Field \[Seite 201\]](#)
- [Continue \(Function\) \[Seite 540\]](#)
- [Control Elements - Overview \[Seite 168\]](#)
- [First Page, Last Page \(Function\) \[Seite 505\]](#)
- [Exit or End <application> \(Function\) \[Seite 492\]](#)
- [Help \(Function\) \[Seite 509\]](#)
- [Possible Entries \(Function\) \[Seite 499\]](#)
- [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#)
- [Previous Page, Next Page \(Functions\) \[Seite 535\]](#)
- [Pushbuttons With No Fixed Position \(Control Element\) \[Seite 204\]](#)
- [Save or Post \(Function\) \[Seite 524\]](#)
- [Status Bar \[Seite 305\]](#)

Tab strips

Tab strips

Tab strips allow you to easily and comprehensively define different object components of an application on one screen and to navigate between them. In contrast to the classical screen change, the user at one glance sees all destinations he or she must call to accomplish the given task. Using this technique, the user can comprehend the structure of an application more intuitively as with conventional techniques such as the *Goto* menu. This reduces the learning expense and facilitates the usage.

For more information on tab strips, see the following sections.

[Using a tab strip \[Seite 315\]](#)

[Tab strip components \[Seite 317\]](#)

[Programming tab strips \[Seite 319\]](#)

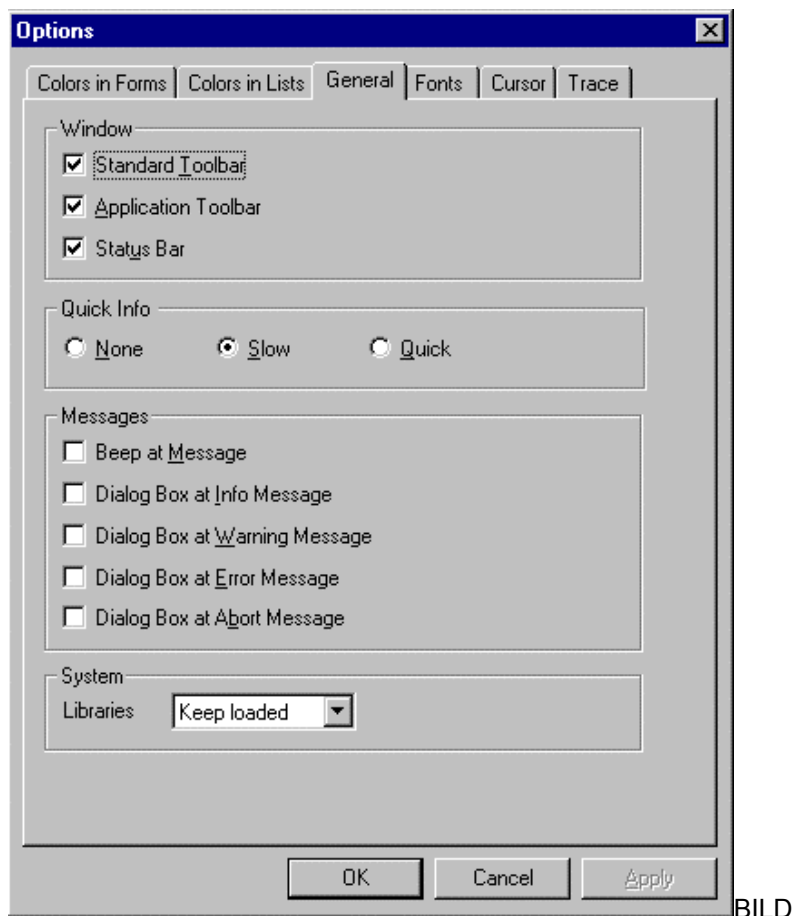
[Defining a tab strip in the Screen Painter \[Seite 321\]](#)

[Further Notes \[Extern\]](#)

Using a tab strip

You use a tab strip to

- structure complex applications and show the navigation supported within. Use a tab strip whenever different object components or logical views for one object exist for which navigation covers more than one screen. Such applications are usually characterized by constant screen areas such as header data and changing contents in the lower screen areas (see Fig. 1).
- present large screens and the navigation between the boxes in a different way.
- display property sheets showing attributes of complex objects or applications. Usually, property sheets are displayed in dialog boxes.



You **must not** use a tab strip if

- the use of the object components is designed as a stand-alone application or if the tab strip environment (menus, pushbuttons, header data, and so on) cannot be kept constant.
- a certain processing sequence is required. A tab strip allows the user free navigation among all object components which contradicts a mandatory sequence.

Using a tab strip

- the object components can be processed dynamically. This is the case if, for example, entering a value on one tab page unexpectedly produces new tab titles.

Tab strip components

Tab titles

Keep the tab titles as short and meaningful as possible. Sort them according to the work flow from left to right.

As the advantage of tab strips consists of displaying all navigation possibilities at once, try not to define more tab titles than can be displayed. However, due to the complexity of R/3 transaction, this is not always possible.

If you define more tab titles than can be displayed, the user must scroll. For this purpose, the system automatically displays scroll buttons for horizontal scrolling at the upper right margin. To get a complete overview of all navigation targets, the user can call a list of all tab titles from a dialog menu (see Fig. 1).

The R/3 System does not support multiple-line tab titles. If multiple lines were used instead of the scroll buttons, due to the language dependency of the titles a different number of tab title lines would appear for each logon language. Since the tab page is of fixed size, the developer would have to save as many blank lines above the tab strip as might be needed under worst conditions. This would waste screen space and reduce readability.

If tab titles correspond to entries in the *Goto* or *Options* menus, delete these entries from the menus. The user must always be able to use the *Goto* menu to reach the screen that contains the tab strip.

For the duration of one processing session, the number as well as the contents of the tab titles should be the same.

The tab strip allows you to define icons as tab titles, however, only in combination with texts. If you used icons only, this would nullify the advantages of a transparent application structure stated above.

Tab page

The Screen Painter locks the first line of a tab page, which the system needs to draw the tab titles. Otherwise, the layout rules for creating screens apply.

A tab strip requires three additional screen lines for the tab titles line, the lower border, and the locked first line.

A tab page together with the tab titles texts corresponds to a group box. Therefore, you need no further box that includes all fields of a tab page.

When converting: Fields that are grouped on old screens using group boxes receive no more box; use the old box title as tab title of the tab page.

However, you can use several group boxes on one tab page. In this case, make sure that the tab title contains a meaningful description of the groups of fields combined on the tab page.

Environment of the tab strip

You must provide a constant environment of the tab strip. This means that if you go from one tab page to another, the menu bar and the application bar must not change. For special functions

Tab strip components

that are active only on one tab page, you must, therefore, use pushbuttons, which you can position anywhere on the respective page. These functions must not appear in the menu.

Leave a blank line between the header data and the tab strip, as usual.

Navigation

The user perceives a screen that contains a tab strip as a unit. The individual tab pages thus are components of the screen, such as boxes of groups of fields. The function keys F3, F12, and F15 therefore do not refer to individual tab pages but to the underlying screen.

If the user navigates from a tab page to another screen, make sure to reactivate the previously active tab page when the user returns to the screen.

Programming tab strips

To create a tab strip in the R/3 System, make sure to meet the following prerequisites:

- GUI version 4.0 and higher
- frontend: Motif, Windows 95, MacOs, NT 3.51 and higher

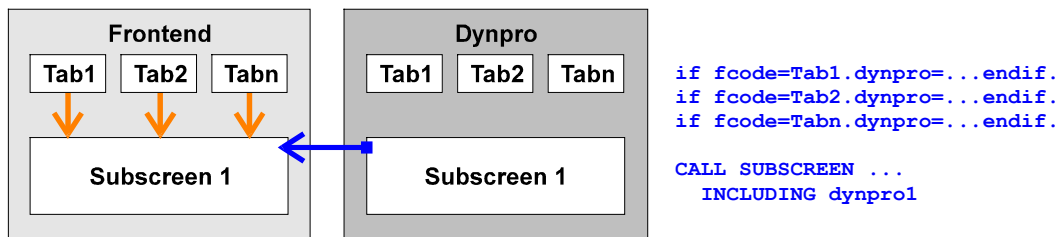
Scrolling in the tab strip

The GUI element *Tab strip* offers two alternatives for scrolling. In both cases, the tab titles are defined as pushbuttons and the tab page as subscreen. However, in one case, scrolling is achieved via the backend (the application server), in the other locally in the GUI.

Scrolling on the application server

Each tab title is assigned the same subscreen area which is embedded at PBO time. When a tab title is chosen, the PAI event is triggered, and the ABAP flow logic must activate the associated tab title page based on the function code, and assign the correct subscreen screen to the subscreen area.

Since each time a tab title is chosen, the PAI event is triggered, workload is placed on the application server. Any entry checks, however, are only carried out for the current tab strip page.



Scrolling in the SAP GUI

Define an individual subscreen area for each tab title. Define the function codes of the titles using the type "P tab strip code"

Programming tab strips

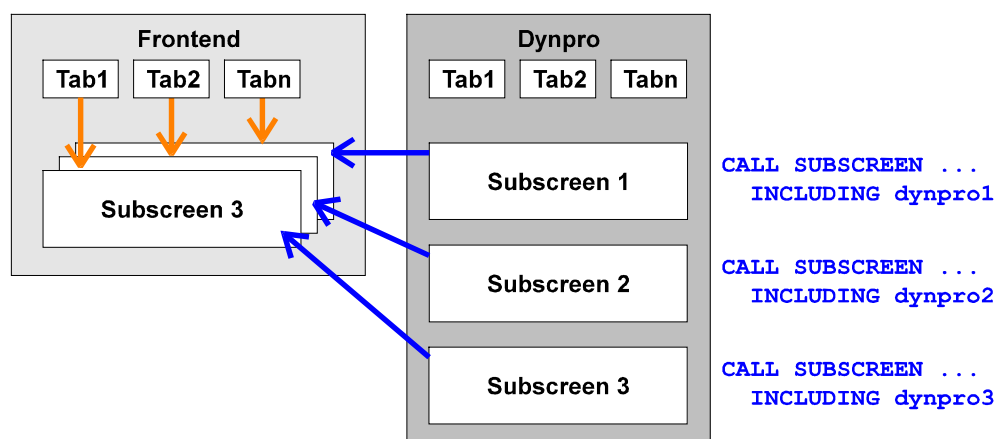
Line	1	Column	2	Length	9	Vis.len.	9
FctCode	TAB2			FctType		Height	1
Groups	Type of function code for pushbutton						
Dict							
Format							
Frm Dict							
Modific.							
ConvExit							
Param.ID							
<input type="checkbox"/> SET param.							
<input type="checkbox"/> GET param.							
<input type="checkbox"/> Up./lower							
<input type="checkbox"/> W/o template							
<input type="checkbox"/> Foreign key							

Type	Short text
E	Exit command (MODULE xxx AT EXIT-COMMAND)
S	System function
T	Call a transaction
	Normal application function
P	Local GUI functions
H	Internal use

In the flow logic, include all subscreens used into PBO. This ensures that all tab pages exist locally in the GUI.

If the user chooses a tab title, the system scrolls between the tab strip pages in the SAP GUI. The entire tab strip behaves like a large screen. In particular, when a tab title is chosen, the system does not trigger the PAI event and does not transport any data. On the one hand, this enhances the performance of the tab strip, but on the other hand all entry checks of the entire screen take place in the case of user actions that trigger the PAI event. If the user performs an action on a tab strip page, it may be the case that the system goes to a required field on another tab strip page that has not been filled.

This procedure is therefore better suited for screens that display data and not for screens that accept user entries.



Defining a tab strip in the Screen Painter

To define a tab strip at the current development stage, you must use the graphical Screen Painter.

Execute the following steps to create a simple tab strip:

1. Set the tab strip borders.
2. Define the tab titles.
3. Define and allocate the subscreen area.
4. Program the flow logic.

Set the tab strip borders

To determine the tab strip area, proceed as follows:

1. Click on the tab strip icon in the elements palette of the Screen Painter.
2. Determine the area by dragging the mouse over the screen.
3. Specify the field name for the new screen element. The field name is the area name you use in the ABAP statement CONTROLS.

Define the tab titles

By default, the new tab strip contains two tab titles. Technically, tab titles are treated like pushbuttons. This becomes obvious when you double-click on a tab title in the attributes window. For each tab title, enter field name, field text, and function code.

You can also define tab titles as templates and fill them with texts at runtime.

To create additional tab titles, drag pushbuttons into the tab titles line.

Define the subscreen area

Each tab title must be allocated to a subscreen area.

Proceed as follows:

1. Select one or more tab titles.
2. Click on the subscreen icon.
3. Define the subscreen area within the tab strip by dragging the mouse. Caution: the first line of the tab strip is locked.

You can also allocate the subscreen by entering the name of the subscreen area into the reference field in the attributes window or in the field list.



For information on creating tab strips in the alphanumeric Screen Painter see [Defining tab strips \[Extern\]](#).

Definition in ABAP

You declare the tab strip in the module pool like Table Controls using the CONTROLS statement.

Defining a tab strip in the Screen Painter

CONTROLS: MYTABSTRIP TYPE TABSTRIP.

In this example, MYTABSTRIP is the field name of the tab strip in the Screen Painter. At runtime, at present only the attribute ACTIVETAB is available, which determines the currently active tab page. You fill it with the function code of the corresponding tab title.

To activate the tab title whose function code is "F1", use the statement

MYTABSTRIP-ACTIVETAB = 'F1'.



The tab strip controlling is fully integrated into the screen concept, allowing you to use it for batch input.

Program the flow logic

Scrolling on the application server



```
DATA: DYNPRNR(4) TYPE C,
      MODUL(30) TYPE C VALUE 'PROGRAMMNAME',
      OK_CODE LIKE SY-UCOMM
```

....

```
CONTROLS: TABSTRIP1 TYPE TABSTRIP.
```

...

Flow logic:

```
PROCESS BEFORE OUTPUT.
  MODULE SETPAGE.
  MODULE MODIFYSCREEN.
  CALL SUBSCREEN SUB1 INCLUDING MODUL DYNPRNR.
```

...

```
PROCESS AFTER INPUT.
  CALL SUBSCREEN SUB1.
  MODULE FCODE.
```

...

```
*-----*
```

```
MODULE SETPAGE OUTPUT.
  IF DYNPRNR IS INITIAL.
    DYNPRNR = '0200'.
    TABSTRIP1-ACTIVETAB = 'TAB1'.
  ENDIF.
ENDMODULE.
```

```
*-----*
```

```
MODULE MODIFYSCREEN OUTPUT.
  LOOP AT SCREEN.
```

...

Defining a tab strip in the Screen Painter

```

        MODIFY SCREEN.
    ENDLOOP.

*-----*
MODULE FCODE.
    CASE OK_CODE.
        WHEN 'TAB1'.
            DYNPRONR = '0200'.
            TABSTRIP1-ACTIVETAB = OK_CODE.
            CLEAR OK-CODE.
        WHEN 'TAB2'.
            DYNPRONR = '0300'.
            TABSTRIP1-ACTIVETAB = OK_CODE.
        ....
    
```

Scrolling in the SAP GUI



Ablauflogik:

```

PROCESS BEFORE OUTPUT.

    CALL SUBSCREEN sub1 INCLUDING 'modul1' '0100'.
    CALL SUBSCREEN sub2 INCLUDING 'modul2' '0101'.
    CALL SUBSCREEN sub3 INCLUDING 'modul3' '0102'.
    ...

PROCESS AFTER INPUT.

    CALL SUBSCREEN sub1.
    CALL SUBSCREEN sub2.
    CALL SUBSCREEN sub3.
    ...
    
```



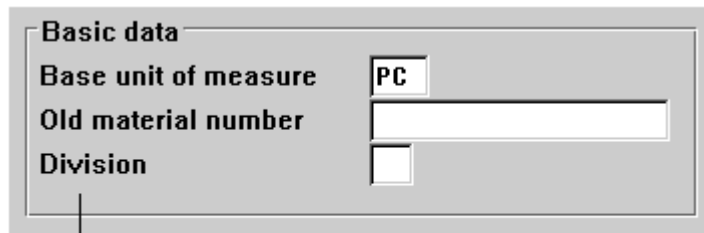
For more information on programming tab strips, see the [Tab Strips \[Extern\]](#) in the ABAP documentation.

Field Names (Control Element)

Field Names (Control Element)

Field names identify the contents of the corresponding input/output fields.

Illustration



The illustration shows a rectangular control element with a light gray background. Inside, the text 'Basic data' is at the top left. Below it, there are four labels: 'Base unit of measure', 'Old material number', and 'Division'. To the right of 'Base unit of measure' is a small input field containing the text 'PC'. To the right of 'Old material number' is a larger, empty input field. To the right of 'Division' is a small, empty input field. A thin vertical line extends from the bottom of the 'Division' label down to the text 'Field name' below the control element.

Field name

Field names as part of the screen

[Use and Appearance \[Seite 325\]](#)

[Design Guidelines \[Seite 326\]](#)

[Icons as Field Names \[Seite 328\]](#)

[Technical Notes \[Seite 329\]](#)

[Cross-References \[Seite 330\]](#)

Use and Appearance

Use

A field name is always assigned directly to an input or output field and briefly describes the contents of this field.

Appearance

Field names appear in two-dimensional, proportional typeface on the template.

All field names are displayed in the same typeface. The character font can be set by the user and applies to all field names.

Design Guidelines

Design Guidelines

Using Familiar Terms

Use words which are part of the user's working vocabulary.

- The German suffix "-nummer" can be dropped, except in the word "Personalnummer" (use "Lieferant" and not "Lieferantennummer").
- Suffixes such as "-schlüssel" and "-kennzeichen" should only be used if they are necessary to make the field intelligible.

Uppercase

Always capitalize the first letter of a field name.

Good:	<i>Initial value</i>	12 , 6	
		<i>Further value</i>	13 , 7
Bad:	<i>initial value</i>	12 , 6	
		<i>further value</i>	13 , 7

Word Order

If a field name consists of two or more words, the part containing the important information should appear at the beginning.

Good:	<i>Create customer</i>	Arthur Miller
Bad:	<i>Customer creation</i>	Arthur Miller

Length

A field name should be no longer than 10 to 15 characters. If the field has been used very often, you can abbreviate the field name.

Exception: First, define an appropriate name for a given field, without paying attention to the length. Then, try to observe the recommended length by using abbreviations (see [Abbreviations \[Seite 593\]](#)). If a field can not be abbreviated, use the original formulation.

Number of Abbreviations

Do not use more than two or three "[abbreviations \[Seite 593\]](#)" in a field name.

Spacing Between Field Name and I/O Field

Leave at least two blanks between the field name and the I/O field. There are no leading characters between field name and input field.

Good:	<i>Customer name</i> Johnson
Bad:	<i>Customer name</i> Johnson

Icons as Field Names

Illustration

The illustration shows a SAP form titled "Vendor". It contains several input fields with different labeling styles:

- Name:** A single-line text input field.
- Address:** Three stacked text input fields, each preceded by a small house icon.
- Delivery date:** A date input field preceded by a calendar icon.
- Order date:** A date input field preceded by a calendar icon.







Field names implemented as text, icon, and icon plus text label

Use

Icons can represent field names (keywords). In general, they replace common terms such as telephone, fax, date or time.

As an exception icons can complete field names. In this case, you should use either an icon plus textual label (for example, address icon plus text "Shipping address" in order to distinguish from "Factory address") or position the icon next to field names in order to indicate a relationship (for example, place the *Copy* icon next to the field names *from* and *to*).

Appearance

	PHONE	Phone number		TIME	Time
	FAX	Fax number		DATE	Date
	Mail	Send mail		ALARM	Alarm

Icons that may be used as field names or as part of field names

Alignment

The rules as specified for field names apply.

Technical Notes

Appearance and Screen Painter Attributes

Field names appear flat and in proportional typeface on the screen template. For dynamic field names, the flags "A" and "FixA" are to be set in the Screen Painter. This also applies to the field names of checkboxes and radio buttons.

Possible Lengths in the ABAP/4 Dictionary

Field names are defined in the ABAP/4 Dictionary. Field names of different lengths are distinguished:

- Short keywords, length 10 (field name)
- Medium-length keywords, length 15 (field name)
- Long keywords, length 20 (field name)
- Header, maximum length 55 (table headings)

Always Leave Two Blanks

Field names to be placed before input/output fields are entered under "keywords". The words entered should always be two characters shorter than the displayed or specified length. The recommended lengths should be used wherever possible in order to facilitate the translation.

Exception: The difference between the number of characters and the entered length may be smaller than two if the full word can then be used and if a meaningful abbreviation by two characters is not possible (this will most often be the case with ten-character keywords).

Icons

Icons which are used as field names are defined in the same way as keywords (text). They are defined as one line and two characters wide.



For a list of the Screen Painter attributes, refer to the section [Input/Output Fields \(Control Elements\)](#) [Seite 238].

Cross-References

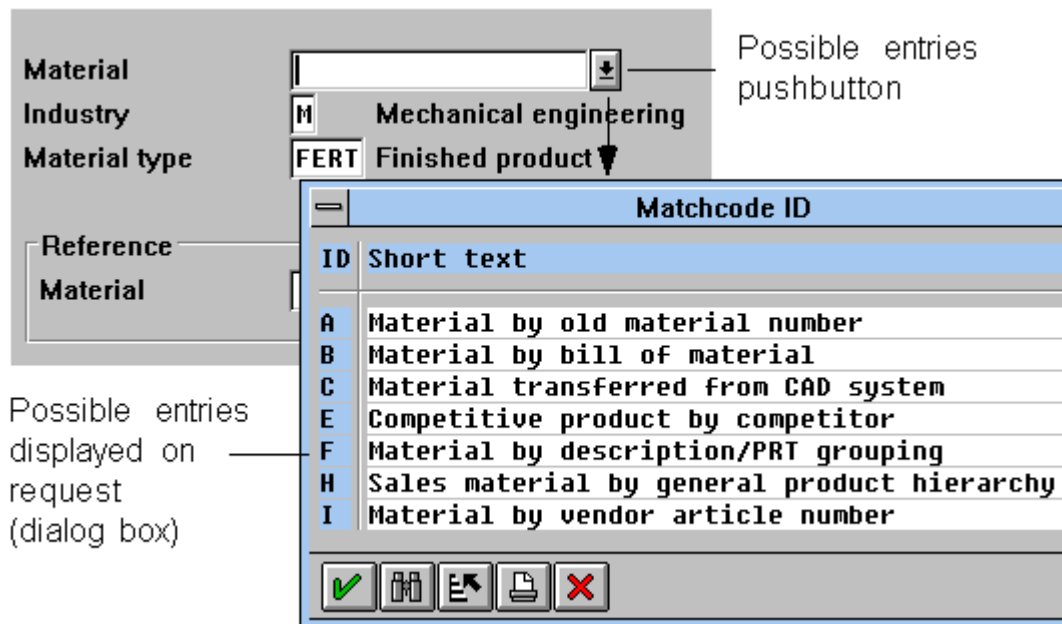
Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Formulations \[Seite 601\]](#)
- [Highlighting \[Seite 603\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)
- [Translation \[Seite 621\]](#)
- [Uppercase/Lowercase \[Seite 619\]](#)

Possible Entries Pushbutton (Control Element)

The Possible entries pushbutton appears directly to the right of the input/output fields which support input help (see [Possible Entries \(Function\) \[Seite 499\]](#)). The pushbutton can either be permanently displayed or only appear if the active control indicator is on the input field.

Illustration



Possible entries pushbutton

[Use \[Seite 332\]](#)

[Effect on the Screen design \[Seite 335\]](#)

[Technical Notes \[Seite 337\]](#)

[Cross-References \[Seite 338\]](#)

Use

Use

Area of Application

The user can display the Possible entries pushbutton for a field by

1. pressing the F4 key when the active control indicator is positioned on the input field
2. pressing the Possible entries pushbutton of the field (for not permanently displayed possible entries pushbuttons, the active control indicator has to be positioned on the input field)

This gives the user two options at present for calling possible entries with the mouse.

The following guidelines discuss the Possible entries pushbutton.



The "Possible entries" pushbutton will be removed from the standard toolbar as soon as all possible entries can be displayed with the Possible entries pushbutton of the individual field.

Using the Possible Entries Pushbutton

As a rule, Possible entries pushbuttons should be provided for all fields which support possible entries help (see the Technical Notes).

On the other hand, Possible entries pushbuttons must **not** be implemented if no possible entries help is available (see the Technical Notes).

Permanent vs. not Permanent Display

There are **permanently displayed** and **not permanently displayed** Possible entries pushbuttons. Not permanently displayed possible entries pushbuttons appear only, if the active control indicator is positioned on the respective field.

Not Permanently Displayed Possible Entries Pushbuttons

For the regular F4 input support, use the **not permanently displayed** pushbuttons.

Justification:

When a template is fully exploited, there are some 20 fields on the screen. About half of these fields support input help. These buttons take up some 30 characters of the screen if they are permanently displayed. Thus, they increase the occupied part of the screen - keep the occupied part as small as possible for the sake of legibility. The pushbuttons also slow down the recognition of field value and check text by reducing the separating space. Finally, they draw the attention to themselves and distract from the important information.

Initial Screens

Not permanently displayed Possible entries pushbuttons are also to be used for initial screens or very empty screens.

Justification:

The user cannot understand why the Possible entries pushbutton is permanently displayed in one case and not in the other. Furthermore, matchcode fields only have a not permanently

displayed pushbuttons nowadays so that they can be used in step loops. But there are many matchcode fields on the initial screen and they would have a permanently displayed pushbutton.

Permanently Displayed Possible Entries Pushbuttons

Permanently displayed Possible entries pushbuttons may only be used if the following requirements are met:

- there is a useful default value
- there are only few possible values (< 20)
- the user changes the default value only rarely
- the fields are control fields with display of free text and not data entry screens.

Justification:

The conditions referred to are displayed in other user interfaces by a so-called pop-up menu, which is not yet available for SAP. The arrangement described here is the closest to a display in a pop-up.

Display Mode

The same guidelines are valid in the display mode. Not permanently displayed Possible entries pushbuttons can be called in the display mode either with the mouse or the active control indicator (CTRL arrow key).

ID	Short text
A	Customers (general)
C	Customers (by class)
E	Customers (export)
K	Customers per account group
S	Customers per sales group
W	?
Z	Customers (head office(s))

Criteria for permanently displayed Possible entries pushbuttons

Use

The following table sums up the guidelines:

	No possible entries pushbutton available	Possible entries pushbutton available
I/O field	No possible entries pushbutton	Not permanently displayed possible entries pushbutton
Output field	No possible entries pushbutton	Not permanently displayed or permanently displayed possible entries pushbutton

Effect on the Screen design

Other screen objects must not be obscured by the Possible entries pushbuttons. **Check texts**, **group boxes** and **right window borders** are very often affected by this guideline.

For **not permanently displayed** Possible entries pushbuttons, **three** blanks are to be inserted between input/output field and check text to the right so that the displayed text is not obscured by the Possible entries pushbutton.



For **permanently displayed** Possible entries pushbuttons, it is not possible in the Screen Painter to fill the first three characters after a field.

Check Texts

Check texts (explanatory texts) should not be aligned if this moves them too far away from the input/output field. For this reason, the check text should generally begin one character after the Possible entries pushbutton.

Dialog Box

If dialog boxes contain a field with a Possible entries pushbutton at the right border, the right window border is to be moved to the right by **three** characters.

Group Box

The right border of a group box with Possible entries pushbuttons can be defined with a minimum spacing of **three** blanks to the longest field with a Possible entries pushbutton.

Exceptions: I/O Fields in Step Loops and Related I/O Fields

I/O fields constitute an exception to this rule. I/O fields in step loops are **not** moved three characters to the right. Related fields (for example, Value and Quantity unit or Date and Date format) should be arranged in direct succession. However, it must still be possible to click with the mouse on the field following to the right, even if the Possible entries pushbutton of the left field is displayed.

The following figures illustrate these guidelines:

Effect on the Screen design

Heading

Field 1: 1234 Short text Field 5: 34566545576
 Field 2: 56754 Short text Field 6: 456345234
 Field 3: 3334 Short text
 Field 4: 1234,00 1b

Heading

Column 1	Column 2	Column 3	Column 4
<input checked="" type="checkbox"/> 123			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Entry 1 of 124

No blanks between related I/O fields

Leave at least 3 characters spacing between I/O field and short text or check text

No blanks in step-loops

Leave at least 3 characters spacing between I/O field and border of group

Effects of the Possible entries pushbutton on the screen design in the primary window

Organizational Levels/Profiles

Organizational levels

Plant: 0001 Werk 0001

Storage loc.: [dropdown]

Data Select view X

Leave at least 3 characters spacing between I/O field and right border of dialog box

Effects of the Possible entries pushbutton on the screen design in the dialog box

Technical Notes

Availability of the Possible Entries Pushbutton

Possible entries help is displayed "automatically" in the SAP System for

- matchcode fields
- fields which are affiliated with check tables
- fields for which domain values or screen values are defined

In addition to this, separate Possible entries can be programmed (in screens via PROCESS ON VALUE REQUEST, in lists by assigning a function code to the (F4) key).

Activating the Possible Entries Pushbutton

The Possible entries pushbutton is displayed if the developer sets the following flag to a field in the Screen Painter, depending on the type of the Screen Painter and the required display of the key:

Screen Painter	Not permanently displayed key	Permanently displayed key
alphanumeric SP	Possible entries pushbutton	permanently displayed possible entries pushbutton
Motif SP	Possible entries pushbutton	permanently displayed possible entries pushbutton

Currently, the flag "possible entries pushbutton" is set automatically for matchcode fields and cannot be switched off.

Work is currently in process for a method of setting the flag automatically when generating the screen.

In future, the flag can be set at runtime.

Cross-References

Cross-References

- [Control Elements - Overview \[Seite 168\]](#)
- [Possible Entries \(Function\) \[Seite 499\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Pushbuttons With No Fixed Position \(Control Element\) \[Seite 204\]](#)

Tables and Lists

This section discusses the use and design of tables and lists. "Tables" here refers to lists of elements in which the user can scroll and edit data. Lists are primarily used for data analysis.

Up to now tables have been implemented using the step loop technique. For new developments, you should use the new table control feature, if possible. Lists are generated by the ABAP/4 list processor.

[Table Control \[Seite 340\]](#)

[Entry and Display Tables \[Seite 355\]](#)

[Arranging Table Fields \[Seite 368\]](#)

[Selection Tables and Lists \[Seite 362\]](#)

[Tables - Scrolling \[Seite 376\]](#)

[Tables - Maintenance \[Seite 383\]](#)

[Lists - General Issues \[Seite 394\]](#)

[Lists - Colors and Lines \[Seite 401\]](#)

[Lists - Hotspots \[Seite 429\]](#)

[Lists - Symbols and Icons \[Seite 432\]](#)

Table Control

Table Control

The table control feature allows to enter and display interactively one-line tabular data structures in an efficient and ergonomic manner. This feature is to replace the step loop technique.

[Functionality \[Seite 341\]](#)

[Description of the Options \[Seite 343\]](#)

[Use and Effect of the Options \[Seite 345\]](#)

[Appearance \[Seite 350\]](#)

[Technical Notes \[Seite 353\]](#)

[Cross-References \[Seite 354\]](#)



Multiple-line tables are still implemented using the step loop technique (For more information, please refer to [Entry and Display Tables \[Seite 355\]](#)).

Functionality

Included Functions

The following functions are supported by table controls and do not have to be implemented by the developer:

Selection

- Inserting selection column and displaying selection state (highlighting).
- Selection functions:
 - *Select/Deselect* cell
 - *Select/Deselect* row/column
 - *Select/Deselect* All

Scrolling

- Scroll functions (horizontal and vertical) can be accessed via the table's scrollbars and via scroll buttons.
- Hiding the vertical scrollbar at runtime, when the visible data area is smaller than the visible table area.

Resizing (vertical)

- Static and/or dynamic resizing of table columns.

Operations on data columns

- Column width can be manipulated directly (the column width is stored independent of the field width).
- Fixation and display of an arbitrary number of key columns.
- Moving columns via drag-and-drop.
- Dynamic hiding of columns by the task.

User settings

- Storing and retrieving user settings for the column layout.
- Disabling (at design time) and thus deactivating (at runtime) the customize pushbutton.

Display

- Displaying column headings as pushbuttons (feature can also be set at runtime).
- Optional display of horizontal und vertical separator lines in the list body.

Displayable Field Types

- Assigning attributes to any cell at runtime.

Functionality

- In the table body, you can use the following controls: scrollbale I/O fields, vertical and horizontal radio buttons, checkboxes.

Not Available or Delayed Functions

- Using pushbuttons in the list body.
- Multiple-line display, especially for text fields.
- Pushbuttons for multiple-line headings.
- Joint colums (i.e. jointly movable columns).
- Ruler (for rows).
- Selecting any number of cells (e.g., not contiguous selections).
- Local display of entry number and current row number.
- Pushbuttons for selections that can be filled with text and icons by the task.
- Horizontal resizing of the table control.

Functions to be Implemented by the Developer

- Inserting blank lines between already filled rows (at the end of the data rows blank lines are inserted automatically).
- Deleting and processing selected objects.
- Calling detail information for selected objects.
- Other task-specific functions.

Pushbuttons for Self-Defined Functions

Arrange the above-mentioned functions as pushbuttons with no fixed position. Place them horizontally and left-aligned below the table control in the following order:

- Detail, Insert line(s), Delete line(s), other functions

Use the appropriate icons for these functions.

Alternatively, you may arrange these functions to the right of the table control, starting at the lower border of the table.

Do not position the general clipboard functions [Cut \[Seite 486\]](#), [Copy \[Seite 513\]](#), and [Paste \[Seite 502\]](#) as pushbuttons at the table control. They are not limited to the table control, but take effect on all selected screen objects. You should therefore place them in the [Standard Toolbar \[Seite 307\]](#).

Description of the Options

Attribute *Table type*: Field *Entry table*, Field *Selection table*

This attribute allows you to specify different table types. The selection determines the background color and the selection logic of the displayed table. Currently, you can choose *Entry table* or *Selection table*.

Attribute *Configurability*: Field *Configurability*

This attribute controls if the user can store changes to the column sequence and width.

Attribute *Resizing (vertical)*: Field *vertical*

This attribute influences the table behavior when the user manually changes the window's size.

Attribute *Separators (horizontal, vertical)* : Field *vertical*, Field *horizontal*

When setting this attribute, the system inserts vertical separator lines between the table lines (field *vertical*) and/or horizontal separator lines between the table columns (field *horizontal*) at runtime.

Attribute *Line selection column*: Field *Line selection column*

This attributes generates an additional column with pushbuttons to the left of the first table field. These pushbuttons locally select a line and replace the previously used checkboxes and radio buttons.

Attribute *Lines (none, single, multiple)*: Field *none*, Field *single*, Field *multiple*

This attribute controls the number of table lines which can be selected at a time:

- If this option is set to *none*, no line selection is possible with the selection buttons.
- If this option is set to *single*, single line selection is possible.
- If this option is set to *multiple*, multiple line selection is possible.

Attribute *Columns (none, single, multiple)*: Field *none*, Field *single*, Field *multiple*

This attribute controls the number of table columns which can be selected at a time:

- If this option is set to *none*, no column selection is possible with the column selection button (column heading).
- If this option is set to *single*, single column selection is possible with the column selection button (column heading).
- If this option is set to *multiple*, multiple column selection is possible with the column selection buttons (column headings).

Description of the Options**Attribute *Possible field types*: Field *Text/Key word***

You use this option to choose the control element which is used to present the field in the corresponding column. Appearance and behavior of the control elements within the table control correspond to their definition for usage outside of tables.

Attribute *Column width*: Field *visLg*

This field displays the width of a table column as defined by the number of Screen Painter columns it contains. It is the same as the visible length of the field and the heading.

Attribute *Fixed Columns*: Field *Fix*

This attribute allows you to exclude columns from being moved with the mouse and during horizontal scrolling.

Use and Effect of the Options

Attribute *Table type: Field Entry table*

Use

You use the *Entry table* table type

- when the table is to be used for entering data.
This primary use is not affected by the modalities of displaying, changing or entering at runtime. From an entry table, you still can select lines or cells to display detail data.
- when the table contains at least one field ready for input.

Effect

When the *Table entry* option is set, the background color of the table fields is the same as the one used in the default, that is, fields ready for input have a white background, display-only fields have the same color as the screen background.

Combining with other attributes

The degree to which these options can be combined with other attributes is restricted as follows:

- Single or multiple line selection requires a line selection column.
- Usually, horizontal and vertical separator lines should be displayed.

Attribute *Table type: Field Selection table*

Use

You use the *Selection table* table type

- when the table is exclusively used to display and to take over entries, or
- the table is exclusively presented in display mode.

Effect

When the *Selection table* option is set, all table fields have a white background. The system does not restrict combination with other attributes. Depending on its use, adhere to the permissible attribute combinations (see Use).

Combining with other attributes

- Set the *Line selection column*
 - when multiple line selection is possible (the *Multiple line selection* attribute is set), or
 - when there are several independent cells (columns) which can be selected, or
 - the line selection should still be visible the next time the application is called.
- Do **not** set the *Line selection column* attribute
 - when only one line at a time can be selected (the *Single line selection* attribute is set), and

Use and Effect of the Options

- when the entire line represents one logical unit, that is, does not consist of individually selectable lines.
- You should not use the *Horizontal/Vertical Separators* attributes.

Attribute *Configurability*: Field *Configurability*

Use

The attribute should be set, if possible. **Exception:** Tables used to control other tables must always have the same structure (for example, in Customizing).

Effect

The attribute is set:

- The button above the vertical scroll bar is active. In the dialog box displayed after the user has pressed the button, the user can save the current column configuration under a name and set a version as the default setting. The user can also select one of the existing versions as the display variant of the current table control.
- The corresponding table control is displayed for the user specified in the stored default setting upon the next call of the program. You can set the table control's context as an additional key for differentiation from the program by setting function group STCX.

The attribute is **not** set:

- The button above the vertical scroll bar is inactive. Therefore, the user cannot save his/her changes to the column layout.

Attribute *Resizing (vertical)*: Field *vertical*

Use

You should always use this attribute

- when pushbuttons for editing the table are located below the table.
- when the table can contain many entries of which the user wants to display as many as possible at a time.

Effect

The attribute is set:

- The space between the last table line and the bottom window frame remains constant when manual changes are made to the window's size. Fields located below the table always remain visible.

The space between these fields and the table remains constant.

- When reducing the window's size, table lines are suppressed, starting with the bottom line. Since all screen fields remain visible, the reduced window does not have a scroll bar. When enlarging the window, the system displays additional lines at the bottom of the table.

The attribute is **not** set:

- The distance between the bottom edge of the table and the bottom window frame is not fixed. When reducing the window's size, screen lines are removed from the visible area, starting

Use and Effect of the Options

from the bottom line. As a result, fields located below the table may not be visible. The window does not get a scroll bar. When enlarging the vertical size of the window, table lines and/or screen fields located below the current visible area are displayed. The system can display as many table lines as defined in the Screen Painter.

Attribute *Separators*: Field *horizontal* (with horizontal separators)

Use

You should use horizontal separator lines only with wide tables which require some sort of visual aid in order to guarantee readability.

You should **not** use horizontal separator lines when

- the table has only one column
- a multiple column table can still be easily read without separators.

Attribute *Separators*: Field *vertical* (with vertical separators)

Use

You use vertical separator lines primarily with entry tables which require separation of the input field boundaries.

Try to separate columns in selection tables by inserting one or two blank characters. You should use vertical separators only if explicit separation of the table columns is required to guarantee readability.

In particular, you should **not** use vertical separator lines

- if the columns are not very wide
- if many columns contain checkboxes, radio buttons or icons.

Attribute *Line selection column*: Field *Line selection column*

Use

You need this option to generate particular types of selection tables.

Effect

When the line is selected, the field of the selection column has the value 'X'.

Combining with other attributes

A selection column is required as follows:

- the *Single* or *Multiple line selection* attributes are set with the *Entry table* option, or
- the *Multiple line selection* attribute is set with the *Selection table* option.

Use and Effect of the Options

Attribute *Lines*: Field *none* (no line selection)

Use

In general, you do not need this option. You can use it to generate an *Entry table* type without a selection column. In this case, the user can enter values, but cannot select lines using the selection buttons.

Combining with other attributes

You **cannot** use this option

- when the *Line selection column* attribute is set.

Attribute *Lines*: Field *single* (single line selection)

Effect

If the user chooses a selection button other than the one for the selected line, the system deselects the old selection and selects the new line.

This option corresponds to the selection behavior of [Radio Buttons \[Seite 193\]](#).

Attribute *Lines*: Field *multiple* (multiple line selection)

Effect

If the user chooses a selection button other than the one for the selected line, the system keeps the current line selection and adds the new line to the number of selected lines.

This option corresponds to the selection behavior of [Checkboxes \[Seite 176\]](#).

Attribute *Columns*: Field *none* (no column selection)

Effect

If this option is set, column selection is not possible with the column selection button (column heading).

Attribute *Columns*: Field *single* (single column selection)

Use

If this option is set, the *Select all* pushbutton is inactive.

Effect

If the user chooses a selection button (heading) other than the one for the selected column, the system deselects the current selection and selects the new column.

This option corresponds to the selection behavior of [Radio Buttons \[Seite 193\]](#).

Attribute *Columns*: Field *multiple* (multiple column selection)

Use

You should use this option when the *Select all* pushbutton is to remain active, that is, when all

lines and all columns should be selectable.

Effect

If the user chooses a selection button (heading) other than the one for the selected column, the system keeps the previous column selection and adds the new column to the number of selected columns.

This option corresponds to the selection behavior of [Checkboxes \[Seite 176\]](#).

Attribute *Possible field types*: Field Text/Key word

Use this option to determine if a table field is to be a normal field or a key word field.

Attribute *Column width*: Field visLg

Use

Increase the visible length of a field, if the user cannot read the column heading properly.

Decrease the visual length, if, far fewer characters have been entered as specified in the defined length. The user should be able to read the heading properly.

Effect

If the column width is smaller than the defined length of the corresponding field, the field becomes a scrollable field.

If the column width is larger than the defined length of the field, the table column is increased by the corresponding number of characters. The number of characters ready for input in the input field is not increased.

In both cases, the heading is displayed in the width of the field's visual length.

Attribute *Fixed columns*: Field Fix

Use

Set this attribute for key fields. In particular, the readability of very wide tables is impaired when, during horizontal scrolling, the attribute fields located to the right cannot be clearly associated with the key field.

Effect

The attribute is set:

- The position of the corresponding column cannot be changed by the user employing the drag-and-drop technique.
- With horizontal scrolling in the table, the column is excluded from the scrolling action and remains visible at the left of the table.

If the attribute is set for a field in column 2 or higher, all columns to the left must also be fixed columns.

Appearance

Appearance

The table control feature allows you to implement different table types. The following figures show some examples:

Entry Tables

Parameter	Reference field/structure	Default value
TEXTLINE1		SPACE
TEXTLINE2	T005-NAME2	
TITEL		
GLOBAL		

Page 1 from 1

An entry table implemented with the table control featrure

User name	Code	TerminalId	Ext-Mode	Int-Mode	Client	
	▽	33	1	1	000	HW14
	▽	32	1	1	000	HW13
	▽	29	1	1	000	HW13
	▽	26	1	1	000	HW12
	▽	23	1	1	000	HW12
	▼	42	1	1	000	P133
	▽	41	1	1	000	P150
	▽	37	1	1	000	HW11
	▽	36	1	1	000	HW12
	▽	35	1	1	000	HW14
	▽	22	1	1	000	P088

An entry table with single-selection for lines implemented with the table control feature

Display Tables

Item	BS	GsB	Account	Description	MW	Tax	Amount	
001	50		0000276000	Cash discount revenue	U1	4,50-	30,00-	↑
002	25		0000000010	Miller and Sons Ltd.		0,00	1.150,00-	
003	40		0000160099	Vendor payables		0,00	0,00	
004	50	0004	0000113101	Bank of Arizona		0,00	1.115,50-	
005	50		0000154000	Input tax	U1	0,00	4,50-	
006	50		0000276000	Cash discount revenue	U1	4,50-	30,00-	
007	25		0000000010	Full Enterprises Co.		0,00	1.150,00-	
008	40		0000160099	Vendor payables		0,00	0,00	
009	50	0004	0000113101	Bank of Arizona		0,00	1.115,50-	
010	50		0000154000	Input tax	U1	0,00	0,45-	
011	50		0000276000	Cash discount revenue	U1	0,45-	30,00-	
012	25		0000000010	Miller and Sons Ltd.		0,00	1.150,00	↓

A display table implemented with the table control feature

Single Selection Table (for Lines)

Personal basic data

→ Tasks	↑
<input type="checkbox"/> Personal data	
<input type="checkbox"/> Bank data	
<input type="checkbox"/> Maternity protection	
<input type="checkbox"/> Org. assignment	
<input type="checkbox"/> Addresses	
<input type="checkbox"/> Family	
<input type="checkbox"/> Contract	
<input type="checkbox"/> Leave entitlement	
<input type="checkbox"/> Time recording	
<input type="checkbox"/> Planned working time	
<input type="checkbox"/> Internal company data	
<input type="checkbox"/> Education and training	↓

Position 1 from 20

A single selection table (lines) implemented with the table control feature

Appearance

Single Selection Table (for Columns)

User name	TCod	Terminal	A
Adam	SE38	p05722	2
FOLLER	FX01	hw1432	1
ZWURGA	AW45	p02312	7
HENRY	QR23	hs1321	4
WORNER	SA38	w04722	1
KRIMLING	TK10	hs3462	3
HOLLAWA	UR32	q02312	6

A single selection table (columns) implemented with the table control feature

Multiple Selection Table (for Lines)

Personal basic data

→	Tasks	↑
□	Personal data	
→	Bank data	
→	Maternity protection	
□	Org. assignment	
□	Addresses	
→	Family	
□	Contract	
□	Leave entitlement	
□	Time recording	
□	Planned working time	
→	Internal company data	↓

Position 1

A multiple selection table (rows) implemented with the table control feature

Technical Notes

The following table shows which table control options you can set dynamically. You have to specify the variables to be set and their values.

Overview of table control fields and options for setting attributes dynamically

Attribute	Field	Dyn.	Variable	Value
Table type	Entry table	---		
Table type	Display table	---		
Configurability	Configurability	---		
Vertical resizing	vertical	---		
Horizontal separator lines	horizontal	yes	mytable-H_GRID	'_', 'X'
Vertical separator lines	vertical	yes	mytable-V_GRID	'_', 'X'
Selection column	Line selection column	yes	mytable-LINE_SELECTOR	'_', 'X'
Line selection	none	yes	mytable-LINE_SEL_MODE	'0'
Line selection	single	yes	mytable-LINE_SEL_MODE	'1'
Line selection	multiple	yes	mytable-LINE_SEL_MODE	'2'
Column selection	none	yes	mytable-COL_SEL_MODE	'0'
Column selection	single	yes	mytable-COL_SEL_MODE *)	'1'
Column selection	multiple	yes	mytable-COL_SEL_MODE *)	'2'
Possible field types	Text/keyword	---		
Column width	visLg	---		
Fixed columns	Fix	yes	mytable-FIXED_COLS **	Number **

Dyn. = can be set dynamically; Number = number of fixed columns; 'X' = option is set

*) If the specific column and/or at least one column is selected, the mytable-COLS_SELECTED variable has the value 'X', otherwise '_'.

**) Value = Number of fixed columns (starting from the left); the dynamic assignment overrules the assignment in the Screen Painter.

Cross-References

Cross-References

- [Arranging Table Fields \[Seite 368\]](#)
- [Copy \(Function\) \[Seite 513\]](#)
- [Cut \(Function\) \[Seite 486\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Paste \(Function\) \[Seite 502\]](#)
- [Selection Methods - Overview \[Seite 95\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)
- [Standard Toolbar \[Seite 307\]](#)
- [Tables - Scrolling \[Seite 376\]](#)
- [Tables - Maintenance \[Seite 383\]](#)

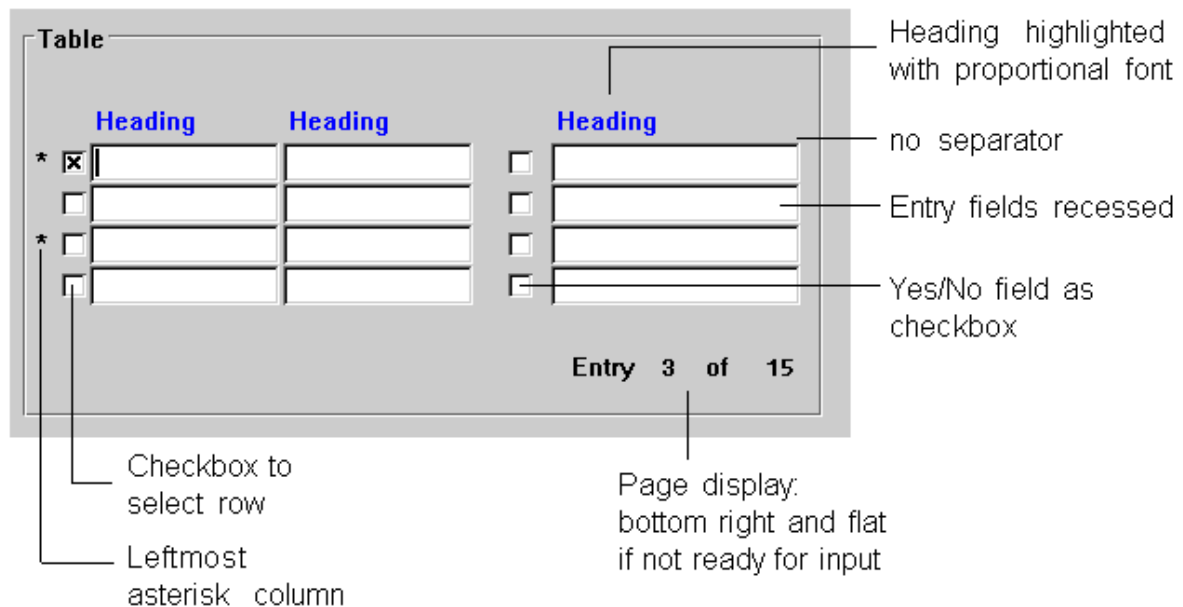
Entry and Display Tables

This keyword discusses the structure and design of entry and display tables. Some of the general design guidelines also apply to selection tables.



This keyword discusses table design via the step loop technique. For new developments please use the [Table Control \[Seite 340\]](#) technique (**exception:** multiple-line tables) and use these guidelines accordingly.

Illustration



[Field Attributes \[Seite 356\]](#)

[Page Display \[Seite 359\]](#)

[Search for Lines \[Seite 360\]](#)

[Cross-References \[Seite 361\]](#)

Field Attributes

Field Attributes

[Abkürzungen \[Seite 593\]](#)

I/O Fields in Entry Tables

For tables in which data can be entered, the input/output fields are always to be displayed **recessed**, even in the display mode.

Entry Table

Fields always recessed, incl. display mode Heading highlighted No separator line, because fields are recessed Inactive checkboxes Unit always recessed

Palletization					Quantities		
SU	Qty per SUnit	SUT	Ty	Sec	Placement qty	Open quantity	Total TO items
<input type="checkbox"/>	X					1	FL
<input type="checkbox"/>	X	1.000				1	0

Items						
Itm	Dest. targ. qty	SUT	Ty	Sec	Dest. bin	Dest. storage unit
1	1.000					
2						
3						

T ☐

I/O Fields in Display Tables

For display-only tables, the input/output fields are to be displayed **flat** on the screen template (flag "FixA"). You should separate the header only for these tables

Field Attributes

Heading highlightend

All fields flat, because never ready for input

Underscore to separate flat entries

Page display flat, because not ready for input

System Messages			
Author	System message		No
WATT	K11 on hs0311 NOT AVAILABLE FOR DIALOG - PLEASE LOGOFF !!!		1
			Line 1 / 1

✓

Display Table

Check Texts

Check texts in entry tables should generally be **recessed** so that the row/column structure is not interrupted.

All Items

It	Material	Order quantity	Un	Short text	X
<input checked="" type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input checked="" type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Leave descriptions or non-changeable fields recessed

Field ready for input: white

Check texts or fields not ready for input: window background color

Check Texts in Entry Tables

Field Attributes

Checkbox and Radio Button Names

Names of checkboxes and radio buttons are always to be displayed **flat** (flag "FixA").

Column Headers

The column headers of tables are to be displayed **highlighted** (flag "Highlighted"), **flat** (flag "FixA") and in **proportional typeface**. If the fields of a table are recessed, do not separate the header and the fields.

Use a blank line inserted between headers and table only for multiple-line tables.

For static column headers, the flag "Highlighted" is to be set in the Screen Painter. For dynamic headings, add the flags "A" and "FixA".

Table heading names are entered under "Header" in the Data Dictionary. The length of the header is specified by the length of the output field.



If you abbreviate column titles, please observe the guidelines in Abbreviations.

Page Display

The page display is to be right-aligned below the table, in the form "Line x of y". "Line", "of" and "y" are always **flat** on the template. If the field "x" is ready for input, it is to be displayed **recessed**; otherwise, it is also displayed **flat**. Due to space reasons, the field "of" should not contain more than three characters, as this is sufficient in most languages.

Do not use leading zeros in the page display.

The diagram illustrates a page display template. It features a table with three columns, each labeled "Heading" in blue. The table has four rows. The first row is marked with a star and a checkbox, and the second row is also marked with a star and a checkbox. The first column contains a star and a checkbox. The first row of the table is highlighted. Below the table, the page display label "Entry 3 of 15" is shown. The word "Entry" is flat, "3" is recessed, "of" is flat, and "15" is flat. Lines connect the labels "Entry", "3", "of", and "15" to the corresponding parts of the page display label. A line also connects the label "Page display bottom right corner" to the "Entry" part. Another line connects the label "Fields flat, if not ready for input" to the "3", "of", and "15" parts.

	Heading	Heading	Heading
* <input checked="" type="checkbox"/>			<input type="checkbox"/>
<input type="checkbox"/>			<input type="checkbox"/>
* <input type="checkbox"/>			<input type="checkbox"/>
<input type="checkbox"/>			<input type="checkbox"/>

Entry 3 of 15

Page display bottom right corner

Fields flat, if not ready for input

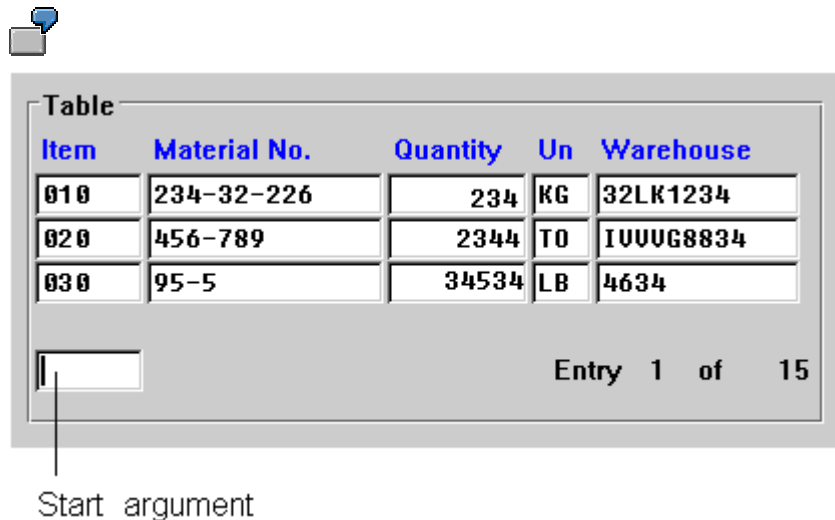
Page Display

Search for Lines

Search for Lines

Set Argument

The input field for a start argument appears in accordance with the column structure below the table to be scanned. The start argument does not receive a field name. If possible, it should be in the same line as the page display.



Table

Item	Material No.	Quantity	Un	Warehouse
010	234-32-226	234	KG	32LK1234
020	456-789	2344	TO	IVVUG8834
030	95-5	34534	LB	4634

Start argument

Entry 1 of 15

Function Position...

Using an alternative to the start argument, a table entry can be found with the *Position...* function, along with further search criteria, if required. The position is entered in the following dialog box. If it is a table in the primary window, the entry *Position...* can be found in the pull-down menu *Edit* or on a pushbutton with no fixed position below the table. In a dialog box, this function is only available on a pushbutton.

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Arranging Table Fields \[Seite 368\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)
- [Table Control \[Seite 340\]](#)
- [Tables - Scrolling \[Seite 376\]](#)
- [Tables - Maintenance \[Seite 383\]](#)

Selection Tables and Lists

Selection Tables and Lists

In selection tables and selection lists the user can select one or more entries. SAP matrices are used for complicated selections.



This keyword discusses the design of selection tables using the step loop technique. For new developments, please use the [Table Control \[Seite 340\]](#) technique and follow these guidelines accordingly.

[Multiple Selection Lists \[Seite 363\]](#)

[Single Selection Lists \[Seite 365\]](#)

[Matrices \[Seite 366\]](#)

[Cross-References \[Seite 367\]](#)

Multiple Selection Lists

Multiple selection lists are used to choose **as many options as desired** from any number of options.

- The list must contain **at least two** options.
- You can select as many lines of the list as you want.



As multiple-selection lists are not an independent element of SAP GUI, they are currently copied by step loops.

Multiple-selection lists can occur independently and appear in a separate dialog box. They can also be a part of primary windows.

Checkboxes

If several lines can be selected at a time in a table, the selection fields are to be displayed as [Checkboxes \(Control Element\) \[Seite 176\]](#) (see Technical Notes).

Checkboxes for selecting table entries are positioned to the left of the respective line, with no space between them and the first field of the table entry (see below).

[Checkboxes \(Control Element\) \[Seite 176\]](#) can also appear in other columns in the table (horizontally, that is to say) and are then to be used as in matrices (see below).

Arrangement

The entries of the selection list are listed one below the other.

If the list exceeds the size of the window, [Scroll Bars \(Control Element\) \[Seite 202\]](#) are to be provided.

Headings

If the list is displayed in a separate dialog box, the guidelines specified under [Title Bar \[Seite 152\]](#) apply with regard to the design of the header line.

Column headers are only used for multiple column tables or for one-column tables, if the title bar does not explain the table contents.

[Checkboxes \(Control Element\) \[Seite 176\]](#) do not have a column header.



Check boxes

Table					
	Item	Material No.	Quantity	Un	Warehouse
<input checked="" type="checkbox"/>	010	234-32-226	234	KG	32LK1234
<input type="checkbox"/>	020	456-789	2344	T0	IUVUG8834
<input type="checkbox"/>	030	95-5	34534	LB	4634

Multiple Selection Lists

Edit Flag

Due to technical reasons, you cannot enter an asterisk "*" into [Checkboxes \(Control Element\) \[Seite 176\]](#) at this time. To do this, a separate column has to be implemented (flag "FixA"). This column should be positioned before the checkbox (see [Selection Methods - Overview \[Seite 95\]](#)).

The screenshot shows a table control with the title "Table". It contains three columns, each with a blue heading "Heading". The first column has a sub-column for an edit flag, indicated by a line pointing to it from the label "Edit Flag" below. The first row of the table is highlighted. The first column contains a star icon and a checkbox. The second column contains a checkbox. The third column contains a checkbox. The status bar at the bottom right of the table indicates "Entry 3 of 15".

	Heading	Heading	Heading
* <input checked="" type="checkbox"/>			<input type="checkbox"/>
<input type="checkbox"/>			<input type="checkbox"/>
* <input type="checkbox"/>			<input type="checkbox"/>
<input type="checkbox"/>			<input type="checkbox"/>

Entry 3 of 15

Edit Flag

The Edit flag appears in a separate column

A selected entry is to be **highlighted** by the task. To achieve immediate highlighting, the F9 functionality is to be supported in addition to the mouse click.

Single Selection Lists

Single-selection lists are used to choose **exactly one** item from any number of options.

- The list must contain **at least two** options.
- **One** option of the list has always to be chosen.



As single-selection lists are not an independent element of SAP GUI, they are currently copied by step loops.

Single-selection lists can occur independently and appear in a separate dialog box. They can also be part of primary windows.

Radio Buttons

For one-column or two-column tables, a [Radio Buttons \(Control Element\) \[Seite 193\]](#) can be placed before the tables to indicate single selection (see Technical notes). For large multiple-column tables, radio buttons should not be used until further notice. The Basis Department is currently working on a solution to this problem.

In addition, the guidelines specified for multiple-selection lists apply.

Matrices

Matrices

If a list of objects exists and several independent options can be set to each object, the objects are to be listed one below the other. The options are set by the user by means of adjacent [Checkboxes \(Control Element\) \[Seite 176\]](#). In this case, the text of the option or the checkbox appears as a column header above the checkboxes. If possible, the checkboxes appear centered below the column headers.

This guideline also applies to [Radio Buttons \(Control Element\) \[Seite 193\]](#). The options of a line must be an affiliated group in the Screen Painter.

Heading	Option 1	Option 2
Object 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Object 2	<input type="checkbox"/>	<input type="checkbox"/>
Object 3	<input type="checkbox"/>	<input type="checkbox"/>
Object 4	<input type="checkbox"/>	<input type="checkbox"/>

Checkboxes in Matrices

Cross-References

- Abkürzungen
- [Erfassungs- und Anzeigetabellen \[Seite 355\]](#)
- [Markiertechniken - Übersicht \[Seite 95\]](#)
- [Tabellen - Pflege \[Seite 383\]](#)
- [Abbreviations \[Seite 593\]](#)
- [Arranging Table Fields \[Seite 368\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Lists - Colors and Lines \[Seite 401\]](#)
- [Lists - General Issues \[Seite 394\]](#)
- [Lists - Hotspots \[Seite 429\]](#)
- [Lists - Symbols and Icons \[Seite 432\]](#)
- Selection Methods - Overview
- Selection Tables and Lists
- [Table Control \[Seite 340\]](#)
- [Tables - Scrolling \[Seite 376\]](#)
- Tables - Maintenance

Arranging Table Fields

Arranging Table Fields

This keyword discusses the arrangement of table fields. The guidelines apply to all types of tables.

ABAP/4 lists for data analysis are dealt with separately. However, some of the general design guidelines also apply to these lists.



This keyword discusses the design of tables using the step loop technique. For new developments, please use the [Table Control \[Seite 340\]](#) technique (**exception:** multiple-line tables) and follow these guidelines accordingly.

[Alignment of Output Fields \[Seite 369\]](#)

[Arranging Column Titles \[Seite 370\]](#)

[General Design of One-Line Tables \[Seite 371\]](#)

[Multiple-Line Tables \[Seite 373\]](#)

[Technical Notes \[Seite 374\]](#)

[Cross-References \[Seite 375\]](#)

Alignment of Output Fields

Alphanumeric fields are left-aligned below one another. Numeric fields, in particular if they are to be compared with each other, are right-aligned below one another.



Good

12123,23	D-2000
323,33	NL-3442
3232,34	B-144

Bad

12123,23	D-2000
323,33	NL-3442
3232,34	B-144

A user finds alphanumeric values by searching through the first letters of a list on the left-hand side; it is for this reason that the left-alignment is to be preferred.

It is easier to compare numeric values and to locate them in a list, if the amount of the value is also indexed by means of the right alignment.

Arranging Column Titles

Arranging Column Titles

If a column title is shorter than the respective table field, the column title is aligned with data of the table. This means in particular that column titles above right-aligned figures should also be right-aligned (see Technical notes).

Exception: If the table is mainly used for the data input, the column title should always be left-aligned.



Good

Names	Numbers
Springfield	1234,56
Baker	1212,23
Westinghouse	12,23

Bad

Names	Numbers
Springfield	1234,56
Baker	1212,23
Westinghouse	12,23

General Design of One-Line Tables

As with block groups, the table should have a heading. If the table appears in a group box, the heading is executed as a group heading.

If the table **basically** displays the contents of the template, you can also include the description of the table in the title bar of the template and convert the header line into a separator. If **only** the table is on the template, you can start with the column titles as of the first line of the work area of the template.

Column Title

For every table column, there must be a column title, unless a particular column is closely related to the one preceding it (as in the case of a quantity unit or currency unit, for example). In such a case, you can omit the column title.

Column Spacing

Position the individual columns of the table right next to one another so that the recessed areas of the table fields form a uniform area.



With Header Line

Table heading				
Item	Material No.	Quantity	Un	Warehouse
010	234-32-226	234	KG	321k1234
020	456-633	2344	TO	10008834
030	95-5	34534	LB	4634

Without Header Line

Document	1234/56	Period	7/95
Date	07/26/1195	Contact	Miller

Item	Material No.	Quantity	Un	Warehouse
010	234-32-226	234	KG	321k1234
020	456-633	2344	TO	10008834
030	95-5	34534	LB	4634

General Design of One-Line Tables

Arranging Columns

Make sure that during the normal filling of the columns with data, as few vertical gaps as possible appear in the table. These gaps can occur because columns whose values can be long but which are generally filled with short values appear in the middle of the table. Try to move these columns over to the right border of the table.

If the user has to compare two columns A and B with a column C, then you can place column C between A and B.



The same columns are to be found in both the good and the bad example, simply with a different arrangement in each case.

Good

Table heading				
Item	Material No.	Quantity	Un	Warehouse
010	234-32-226	234	KG	321k1234
020	456-633	2344	TO	10008834
030	95-5	34534	LB	4634

Bad

Table heading				
Item	Material No.	Quantity	Warehouse	Un
010	234-32-226	234	321k1234	KG
020	456-633	2344	10008834	TO
030	95-5	34534	4634	LB

Multiple-Line Tables

Try to avoid two-line tables

- by transferring these to several one-line tables or
- by providing only one data row, with further data being displayed on a "normal" template by choosing a table entry.

Three-line tables should always be avoided..

Exception: Multiple-line tables are permissible if the lines basically contain long texts.

General Design of Two-Line Tables

The first line of the column titles and the corresponding data rows are **highlighted**. The second lines appear in normal intensity.

Otherwise, the guidelines specified for one-line tables apply.



Table heading				
Item	Material No.	Quantity	Un	Warehouse
Short text	Batch No.	Issue quantity		
010	234-32-225	234	KG	32LK1234
Old boots	345456/12	20000		
020	456-633	2344	TO	IUVUG8834
Trousers	5976/45	456		
030	95-56	34534	KG	4633
T-shirts	3344/88	1646		
Entry 4 of 5				

Technical Notes

Right-Aligning Column Titles

The right-alignment of column titles causes technical problems:

If you maintain the column title in the ABAP/4 Dictionary, you cannot enter it right-aligned (for example, _____ amount). To right-align the column title, you can therefore:

- either display this in a continuous header line in the Screen Painter (although this means that you do not have field documentation from the ABAP/4 Dictionary), or
- make the description as short as the German text and right-align it above the values of the table (although this creates problems with the translation).

As both alternatives are problematic, they should only be used in exceptional circumstances.

Screen Painter Flags

Please refer to [Input/Output Fields \(Control Elements\) \[Seite 238\]](#) for an overview of the Screen Painter flags for table fields.

Technical Restrictions

For Templates in Step Loops

At present, you cannot implement two templates in step loops as a graphical selection field (such as [Checkboxes \(Control Element\) \[Seite 176\]](#) or [Radio Buttons \(Control Element\) \[Seite 193\]](#)) with a field name. Only the graphical selection field can be implemented as a graphical element. As a consequence, the checkbox cannot be activated by clicking the field name.

For Radio Buttons in Step Loops

For step loops, the system does not set any default radio buttons. Furthermore, the task has to set the radio buttons during the scroll action.

Cross-References

- Abkürzungen
- [Abbreviations \[Seite 594\]](#)
- [Selection Tables and Lists \[Seite 362\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Fields - Positioning \[Seite 253\]](#)
- [Selection Methods - Overview \[Seite 95\]](#)
- [Table Control \[Seite 340\]](#)
- [Tables - Scrolling \[Seite 376\]](#)
- [Tables - Maintenance \[Seite 383\]](#)

Tables - Scrolling

Tables - Scrolling

[Navigationsfunktionen - Übersicht \[Seite 460\]](#)

This section describes functions the user can use to scroll through information in table structures. "Table structures" here refers to all lists of elements in which the user can scroll - that is, lists, step loop areas on the screen, and so on.



This keyword discusses the interaction with tables using the step loop technique. For new developments, please use the [Table Control \[Seite 340\]](#) technique (**exception**: multiple-line tables), which includes the scroll logic.

Syntax of Examples

The following examples refer to a table structure with seven lines of which three lines can be displayed at a time. The examples are displayed as follows:

```
123 --NS-> 456 --NS--> 7
```

This means:

The lines 123 are displayed on the screen. The user then chooses the [Previous Page, Next Page \(Functions\) \[Seite 535\]](#). The lines 456 are then displayed. Further scrolling displays line 7 on the screen.

Empty lines ready for input are indicated by an underscore "_".

[Scroll Functions and Scroll Keys for Vertical Scrolling \[Seite 377\]](#)[Horizontal Scrolling \[Seite 379\]](#)[Scroll Bars \[Seite 380\]](#)[Technical Notes \[Seite 381\]](#)[Cross-References \[Seite 382\]](#)

Most of the functions dealt with in this section are described in [Functions - Alphabetical List \[Seite 467\]](#). You find compressed functions lists in [Functions - Overview \[Seite 451\]](#).

Scroll Functions and Scroll Keys for Vertical Scrolling

[Erste Seite \[Seite 505\]](#)

Vertical scrolling is necessary when all the available information of table structures do not fit in a window.

First/Last Page

The [First Page, Last Page \[Seite 505\]](#) function displays the first screen page and the last page of a table structure.

While the *First page* function always goes to the beginning of the table structure, the *Last page* function gets the user to:

- the last "physical" page if it is only possible to scroll page by page in the table structure.



123 --LS--> 7

- the last full page when it is possible to position on individual lines in the table structure.



123 --LS--> 567

Previous/Next Page

[Previous Page \[Seite 535\]](#) scrolls one page back in a table structure, [Next Page \[Seite 535\]](#) one page forwards. No overlapping occurs.

Exception: In the editor display and in text processing the display contains one overlapping line.



456 --VS--> 123 --NS--> 456 --NS--> 7

Remaining on the Last Page

If the user is on the last page and chooses *Next page*, the display remains on the last page.



456 --NS--> 7 --NS--> 7 123 --LS--> 7 --NS--> 7

Consistent Scrolling

If, in a table structure with a key, the user displays a page somewhere "between" actual pages and then chooses [Previous Page \[Seite 535\]](#) and/or [Next Page \[Seite 535\]](#), the system displays the previous and/or next screen page relative to the current page display



123 Display as of 3: 345 --NS--> 67

Scroll Functions and Scroll Keys for Vertical Scrolling

Please note in the following example that the first *Previous page* consistently scrolls relative to the previous page display, the second *Previous page*, however, stops at the beginning of the table structure, because it is not possible to scroll back three complete screen pages.



123 --LS--> 567 --VS--> 234 --VS--> 123

Vertical Cyclical -Scrolling

Cyclical vertical scrolling is not allowed, because it would confuse the user.

Scroll Keys

Using the keyboard, the user scrolls by using the function keys F21–F24, not the ENTER key. The scroll functions are assigned to the following function keys:

•	First page	F21, CTRL-PGUP (on workstations)
•	Previous page	F22, PGUP (on workstations)
•	Next page	F23, PGDN (on workstations)
•	Last page	F24, CTRL-PGDN (on workstations)

Scroll Arrows

In addition to the function keys F21–F24, there are four scroll arrows in the standard toolbar which the user can activate with the mouse

The scroll arrows are removed from the [Standard Toolbar \[Seite 307\]](#) as soon as correct performance of the [Scroll Bars \(Control Element\) \[Seite 202\]](#) is guaranteed.

Horizontal Scrolling

[Erste Seite \[Seite 505\]](#)

Horizontal scrolling can be necessary if table structures are wider than one screen page

Functions for Horizontal Scrolling

You can use two functions to scroll to the right and left, *Previous horizontal page* and *Next horizontal page*. There are no particular function keys reserved for these functions.

Functions that work in the same way as [First Page, Last Page \(Function\) \[Seite 505\]](#) (that is, "All the way to the right" and "All the way to the left") can be implemented, but do not have to be made available.

Horizontal scrolling can refer to a part of the template. Typically, a vertical section of the template remains constant on the left while you scroll through the right-hand side (in the lower part, the line item number can remain constant, for example).

Page Display

Optionally, the page display for horizontal scrolling can be positioned in right justified form below the table structure. If a page display is already there, it can be displayed to the left from it or in a right justified form with the table structure in an empty line between column titles and table structure contents (see example below).

An appropriate description should be used for the various horizontal pages.



Item	CompNo	Quantity	Unit	Warehouse
Short text	batch number		issue quantity	
View 3 of 5				
10 2346326226	234	Kg	32LK1234	
Old boots	345345/12		2000	
etc.				

Branching Horizontally

Consider whether branching to the various "right-hand" parts of the table structure could be called up via user function calls rather than by horizontal scrolling. Every function call should then be given a title which reflects the generic nature of the data displayed.



F5 = Purchasing view F6 = Sales view F7 = Warehouse view

Scroll Bars

Scroll Bars

[Blätterleisten \[Seite 202\]](#)

[Bedienelemente \[Seite 167\]](#)

[Bedienelemente - Übersicht \[Seite 168\]](#)

Vertical and horizontal scrolling in table structures can also be implemented with [Scroll Bars \(Control Element\) \[Seite 202\]](#). Scroll bars are operated with the mouse.

Technical Notes

Funktion Codes for Scroll Functions

The vertical scroll functions described have the following function codes in the ABAP/4 report processing:

•	<i>First page</i>	P--
•	<i>Last page</i>	P++
•	<i>Previous page</i>	P-
•	<i>Next page</i>	P+

Further function codes exist for relative and horizontal scroll functions. These functions are not described here.

Function Modules

The ABAP/4 function library has a number of function modules which support scrolling in reports. The module SCROLLING_IN_TABLE in the function group STAB covers the scroll functions described in this section. Other function modules are in task-specific function groups.

Cross-References

Cross-References

- [Bedienelemente - Übersicht \[Seite 168\]](#)
- [Symbolleiste \[Seite 307\]](#)
- [Arranging Table Fields \[Seite 368\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Functions - Overview \[Seite 451\]](#)
- Input/Output Fields (Control Elements)
- [Lists - Colors and Lines \[Seite 401\]](#)
- [Lists - General Issues \[Seite 394\]](#)
- [Lists - Hotspots \[Seite 429\]](#)
- [Lists - Symbols and Icons \[Seite 432\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Scroll Bars \(Control Element\) \[Seite 202\]](#)
- Selection Methods - Overview
- [Selection Tables and Lists \[Seite 362\]](#)
- [Table Control \[Seite 340\]](#)
- [Tables - Scrolling \[Seite 376\]](#)

Tables - Maintenance

This section describes functions the user can use to maintain table structures. "Table structures" here refers to all lists of elements through which the user can scroll - that is, lists, step loop areas on the screen, and so on.



This keyword discusses the interaction with tables using the step loop technique. For new developments, please use the [Table Control \[Seite 340\]](#) technique (**exception**: multiple-line tables) and employ the following guidelines accordingly.

Syntax of Examples

The following examples use an unsorted screen table structure with four lines. The examples are displayed as follows:

```
1 2 3 4 --Fx--> 1 2 _ 3 --> 1 2 9 3 --ENTER--> 1 2 9 3
```

This means:

The cursor is on line three (underlined). The user then chooses a function Fx with a function key, pushbutton or from a pull-down menu. After the first two lines the system inserts a blank line for a new entry. Finally, the user enters "9" and presses the ENTER key: The new entry is copied and inserted.

[Functions for Inserting and Appending Lines \[Seite 384\]](#)

[Copy, Move, Delete, and Sort Functions \[Seite 387\]](#)

[Working With the Clipboard \[Seite 392\]](#)

[Cross-References \[Seite 393\]](#)



The functions dealt with in this section are described in [Functions - Alphabetical List \[Seite 467\]](#). For more information on selection methods, please refer to [Selection Methods - Overview \[Seite 95\]](#).

Functions for Inserting and Appending Lines

[Zeile einfügen \[Seite 498\]](#)

[Zeile einfügen \[Seite 498\]](#)

[Zeile einfügen \[Seite 498\]](#)

Insert Line

The [Insert Lines \[Seite 520\]](#) function allows the user to insert a blank line into a table structure and to enter a new element.

Procedure

- Move the cursor to the entry **before** which you want to insert a new line.
- Choose the function [Insert Lines \[Seite 520\]](#).
A message in the status bar requests the user to insert data.
- A blank line is inserted above the cursor position.
- Enter the data and then press `ENTER`.
- The system inserts the data into the table structure and displays the present page of the table structure.

If the user does not enter any data, the blank line disappears. If the user enters incorrect data, the system executes an error procedure.



1 2 3 4 --Fx--> 1 2 _ 3 --> 1 2 9 3 -- ENTER --> 1 2 9 3

Inserting With a Dialog Box

Alternatively, the entry can occur in a dialog box. If the dialog box is too small to accommodate all of the user's data, it can be enlarged up to full screen size.

Inserting Several Lines

To insert several lines, the user can choose the *Insert Lines* function repeatedly. The following functions are, however, more efficient.

Insert Many

The function [Insert Many \[Seite 534\]](#) allows to create several new elements anywhere in the table structure. The present screen page is expanded accordingly.

Procedure

- Der Benutzer stellt den Cursor auf den Eintrag, **vor** den neue Einträge eingefügt werden sollen.
- Choose the function [Insert Many \[Seite 534\]](#).

Functions for Inserting and Appending Lines

- The table structure is expanded **above** the selected element. This means, that the line before the cursor position is displayed as the top line of the page while the line on which the cursor was positioned is displayed as the bottom line of the page.

Between the top and bottom lines, the system displays blank lines where the user can enter new data.

- The further processing is similar to the processing of the Insert Lines function, with the following difference:
If further data is to be entered after entering the new data, then instead of the ENTER key, the function [Insert Many \[Seite 534\]](#) can be called again. As a result, another input screen is displayed: The last line of the already entered data is at the top, and the last line of the preceding insert action is again at the bottom.

After pressing ENTER, the empty lines disappear again.



```
1 2 3 4 --Fx--> 2 _ _ 3 --> 2 9 _ 3 --ENTER--> 2 9 3 4
1 2 3 4 --Fx--> 2 _ _ 3 --> 2 9 _ 3 --Fx--> 9 _ _ 3 --> 9 7 6 3
```

New Lines



With Release 4.0, the meaning and usage of the *New entries* function has changed. Consider these changes for your new developments.

If, for a table, you assume that the user wants to enter data (independent of the cursor position), display the table with additional input-enabled lines.

If, for authorization reasons, you must check whether the user is allowed only to change the table, execute this check as soon as the user enters the application, if possible. In this case, display the table without additional input-enabled lines. If the user is authorized to both change and create the table, present the table with additional input-enabled lines.

Only if this is not possible, you may return to the old version, using the *New entries* function (see below).

The function New Lines allows you to create several new elements at the end of a table structure. The word "Lines" can be replaced by a more appropriate term, depending on the task. In contrast to the function [Insert Many \[Seite 534\]](#), this function does not involve cursor positioning.

Procedure

- Choose the function New Lines.
- The last line of the table structure is displayed at the top of an empty screen page.
To get to the empty screen page, the user should choose a function and not use the Scroll function.
A message in the status bar requests the user to enter data now.
- Enter data and then press ENTER
- Table structures which are **not** sorted by field: The new lines are appended to the end of the table structure.

Functions for Inserting and Appending Lines

Table structures which are sorted by field: The new lines are sorted into the table structure.

- Display after appending:

Table structures which are **not** sorted by field: The table structure is displayed from the last line before the action.

Table structures which are sorted by field: The table structure is displayed from the first element inserted.

After pressing `ENTER`, the empty lines disappear again.



Unsorted

1 2 3 4 --Fx--> 4 _ _ _ --> 3 8 5 _ --ENTER--> 3 8 5 _

Copy, Move, Delete, and Sort Functions

The following functions *Move*, *Copy* (so-called direct copying), *Delete*, etc. must be distinguished from those functions that use the clipboard (*Cut*, *Copy*, *Paste*) which are described later in this section. The functions listed here can be described as **modal** functions because they change to a particular mode (for example, the *Copy* mode), in which no other functions can be executed.

Move

The function [Move \[Seite 532\]](#) allows the user to move lines to a new position in a table structure.

Procedure

- Select the lines you want to move.
- Choose the function [Move \[Seite 532\]](#).
- A message in the status bar informs the user that he or she is now in the Move mode. Lines to be moved are highlighted.
- Move the cursor to the line **before** which the selected lines are to be moved and press the **ENTER** key.
- The selected lines are moved.

The first line moved is now displayed as the top entry.



1 2 3 4 --Fx--> 1 2 3 4 --ENTER--> 3 1 2 4

Copy

The direct copy function [Copy \[Seite 513\]](#) is similar to the function Move, except that the copied information also remains unchanged in its original position.

Procedure

- Select the lines you want to copy.
- Choose the function [Copy \[Seite 513\]](#).
- The lines copied are transferred to a separate empty page. There are no further entry lines.
A message in the status bar informs the user that he or she is in the Copy mode.
- Change the copied data of the copies and then press **ENTER**.
- The data is copied and inserted in the table structure. The table structure shows the page where the first copied line is as the first entry.

Table structure which are sorted by term: The copies are sorted.

Table structures which are **not** sorted by term: The copies are displayed after the copied entries.

For repositioning, you must call the function Move.

If the user enters incorrect data, the system executes an error procedure.

Copy, Move, Delete, and Sort Functions



1 2 3 4 --Fx--> 3 --> 9 --ENTER--> 9 4 _ _

Copy With a Dialog Box

Alternatively, you can copy via a dialog box.

Procedure

- Select the lines you want to copy.
- Choose the function [Copy \[Seite 513\]](#).
- The data is displayed line by line in a dialog box.
- Change the data and then press the `ENTER` key.
- The next line of previously selected data is displayed.
- After the last line has been changed, the data is copied. The table structure shows a screen page with the first line copied displayed as the top entry.

Table structures which are sorted by term: The copies are sorted.

Table structures which are **not** sorted by term: The copies are displayed after the copied entries.

For repositioning, you must call the function Move.

If the user enters incorrect data, the system executes an error procedure.

Copying Generically

The generic symbols "*" and "+" can be used to copy generically. The wildcard symbol "*" stands for any number of characters, "+" takes the place of exactly **one** character.



- Copy all RFSCHU* to RLSCHU*
- Copy all Company code 2

Plant 1+

to Company code 4

Plant 3

Copying All Generically Specified Data Records

- Choose the function [Copy \[Seite 513\]](#).
 - If both direct and generic copy functions are available in your task, you should name this function *Mass copy*.
- A dialog box is displayed in which the lines to be copied and the changes to be carried out on the copies can be specified generically (see example above).
- Press the `ENTER` key.

Copy, Move, Delete, and Sort Functions

- If an error occurs during the copy process because the copies already existed, an error procedure is executed.
- The system displays a screen page with the first copy as top entry.
 - Table structures which are sorted by term: The copies are sorted.
 - Table structures which are **not** sorted by term: The copies are displayed after the copied entries.
 - For repositioning, you must call the function Move.

Copying Generically With Subsequent Changes to the Copies and Selection of the Records to be Copied

- Choose the function Copy.
 - If both direct and generic copy functions are available in your task, you should name this function *Mass copy*.
- The system displays a dialog box in which the lines to be copied and the changes to be carried out for the copies can be specified generically (see example above).
- The copies are shown on a new page of the table structure. To change all lines, the user may have to scroll.
- Change the data and select the datasets to be copied. Then press the `ENTER` key.
 - If the user enters incorrect data, the system executes an error procedure.
- The system displays a screen page with the first line copied as the top entry.
 - Table structures which are sorted by term: The copies are sorted.
 - Table structures which are **not** sorted by term: The copies are displayed after the copied entries.
 - For repositioning, you must call the function Move.

Delete Lines

The function [Delete Lines \[Seite 515\]](#) allows the user to delete one or more lines from a table structure.

Procedure

- Select the lines you want to delete.
- Choose the function [Delete Lines \[Seite 515\]](#).
- If important data would be deleted, a confirmation prompt is issued. It is up to the task developer to decide whether data is important. If in doubt, the prompt should be issued.
 - If both important and unimportant data would be deleted, a confirmation prompt must be issued.
- Following the delete action, the line which was the top entry in the table structure is still the top entry. If this first line was deleted, the preceding line is displayed as top entry.

Copy, Move, Delete, and Sort Functions



1 2 3 4 --Fx--> 1 2 4 _

Notes

The function [Delete Lines \[Seite 515\]](#) either affects the selected area or, if no lines were selected, the cursor position, that is, a line before the cursor position is deleted. It is **not** moved to the clipboard.

If **several** lines are selected, a dialog box with a message should be issued. **Diagnosis:** Several lines were selected. **Question:** Do you want to delete the selected lines?

Swap

With the function *Swap*, two lines of a table structure are swapped. This is only possible if the table structure is not sorted by lines.

Procedure

- Select a line.
- Choose the *Swap* function.
- A message in the status bar informs the user that he or she is in the Swap mode.
- The user positions the cursor on another line and then presses `ENTER`.
- The lines are swapped. After the swapping, the first line in the table structure is displayed as the top entry of the screen page.



1 2 3 4 --Fx--> 1 2 3 4 --ENTER--> 3 2 1 4

Sort

The function [Sort \[Seite 527\]](#) allows the user to sort a table structure according to specified criteria. This also allows an existing sort sequence to be changed.

Procedure

- Choose the function [Sort \[Seite 527\]](#).
- A dialog box is displayed in which the user can specify the sort criteria for the table structure. The user can also enter the sort feature sequence into a field (see [Sort \(Function\) \[Seite 527\]](#)).



_ position
2 name
_ city
1 street

- Press the `ENTER` key.

Copy, Move, Delete, and Sort Functions

- The table structure is sorted and displayed in the new order. The first page is displayed.

Working With the Clipboard

Working With the Clipboard

The clipboard is a separate storage area in which a piece of information can be stored temporarily. There are functions that transfer the information to, or retrieve it from the clipboard. These functions can typically be used as an alternative to the Copy and Move functions described above. If both options are available, the clipboard functions should be preferred.

Three commands work with the clipboard:

- **Copy:** This command copies the line at the current cursor position to the clipboard.
- **Cut:** This command moves the line at the current cursor position to the clipboard. This command thus deletes the line from the table structure.
- **Paste:** This command inserts the line stored in the clipboard before the line at the current cursor position. If a data structure which is not compatible with the table structure is to be inserted, an error message is issued.

These three functions are above all useful for transferring data between different table structures.



This functionality should not be confused with the local cut-and-paste functionality offered by the frontend.

Comparison With Modal Functions

Combinations of the three functions result in the "conventional" functions for modifying a table structure:

- **Direct Copy** = Copy + Paste
- **Delete** = Cut (the first contents are lost with repeated cutting or copying into the clipboard)
- **Move** = Cut + Paste (at another position)

An additional function which allows the user to view the contents of the clipboard in a dialog box should be added to these three functions.

Notes

[Copy \[Seite 513\]](#) and [Cut \[Seite 486\]](#) refer to the selected area of the table structure. If no lines were selected, the line on which the cursor is positioned is copied or moved to the clipboard. A message should be displayed in the status bar: n lines are copied (or moved) to the clipboard.

[Paste \[Seite 502\]](#) is always carried out (once) in the line before the cursor position. The contents of the clipboard remain unchanged.

Cross-References

- Löschen bzw. <Objekt> löschen (Funktion)
- [Tabellenfelder - Anordnung \[Seite 368\]](#)
- [Viele einfügen \(Funktion\) \[Seite 534\]](#)
- [Arranging Table Fields \[Seite 368\]](#)
- [Copy \(Function\) \[Seite 513\]](#)
- [Cut \(Function\) \[Seite 527\]](#)
- [Delete or Delete <object> \(Function\) \[Seite 95\]](#)
- [Entry and Display Tables \[Seite 355\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Insert or Insert <object> \(Function\) \[Seite 498\]](#)
- [Insert Many \(Function\) \[Seite 486\]](#)
- [Lists - Colors and Lines \[Seite 401\]](#)
- [Lists - General Issues \[Seite 394\]](#)
- [Lists - Hotspots \[Seite 429\]](#)
- [Lists - Symbols and Icons \[Seite 432\]](#)
- [Move \(Function\) \[Seite 532\]](#)
- [New Lines \(Function\) \[Seite 520\]](#)
- [Paste \(Function\) \[Seite 502\]](#)
- Selection Methods - Overview
- [Selection Tables and Lists \[Seite 362\]](#)
- Sort (Function)
- [Table Control \[Seite 340\]](#)
- [Tables - Scrolling \[Seite 376\]](#)

Lists - General Issues

When designing lists, it is important to present the information in a simple layout which is easy to understand. As an aid, this topic discusses basic data structures and describes ways in which they can be presented. At the end, there is a simple classification scheme which you should be able to use for most lists.

[Data Structures and Forms of Presentation \[Seite 395\]](#)

[Classifying Lists \[Seite 397\]](#)

[Cross-References \[Seite 400\]](#)

Data Structures and Forms of Presentation

[Listen - Symbole und Ikonen \[Seite 432\]](#)

Creating lists means organizing data. The better your lists are organized, the easier the user will understand what the data means. In the following, we discuss some issues that may help you in organizing your lists.

Data may occur and be organized in endless different ways. Ideally, each data structure should have a corresponding form in which it is always presented. However, since data is often complex and can be viewed in such a different number of ways, there must also be a number of different ways in which they can be presented. This is one of the strengths of the R/3 system, but it makes list design harder for you.

Linear Lists: Single-Line and Multiple-Line Lists

Lets begin with a simple data structure! Often data consists of just linear lists where all lines have the same column structure. This is the simplest and most important case. Normally, you display such data in a one-line table or list. There are cases, however, where the list is wider than the screen or window width. You have two options for this problem: (1) You can force the user to scroll the list horizontally or (2) you can transform the one-line list into a two-line or even multiple-line list. In the first case, it is important that the most frequently used data columns appear on the left so that the need for horizontal scrolling is minimized. In addition, you can fix data columns, usually the key columns, so that users have horizontal guidance. There are, however, cases where scrolling is not possible or not appropriate. Here, you "fold" the "logical" lines into several "physical" lines, thus creating a multiple-line list. Such lists usually look very disordered. Often, you cannot use vertical lines for separating the columns, and you can display fewer logical lines simultaneously on one screen. Therefore, it is very important that you design such lists carefully, try to introduce vertical flight lines and, better, avoid three-line or multiple-line lists altogether.

Matrices

Sometimes data is not arranged sequentially but according to two or more aspects or variables. With two variables, you get a rectangular matrix scheme, which is easy to handle (if it is not too wide). With more variables, you are in trouble — the more variables, the more trouble. In this case, you can do the following: (1) You may use nested headings, and/or (2) you may split the matrix into several matrices. The first case may help for up to three variables per axis. It is very likely that your matrix will no longer fit on the screen horizontally. In many cases, it may be better to split the data into several matrices arranged one below the other or on seperate screens (a tab control would come in handy in this case). Splitting data makes it more difficult to compare. Decide on one of these solutions, dependant on what is important for the user!

Hierarchical-sequential Lists

This type of data structure has a horrible name, but is nevertheless important to the R/3 System. It is closely related to multi-dimensional matrices as well as the hierarchical lists discussed below. All these data structures contain some sort of hierarchy which may be displayed either in tabular or free form. A tabular form is preferable if the 'leaves' of the tree are simple linear lists, as is the case with hierarchical-sequential lists. With matrices, the leaves usually consist of one element only, but the tree is "complete", that is, all combinations of variables have to be displayed. This again calls for a tabular design. Trees are more useful, if the hierarchy is irregularly shaped, is partially visible, and involves a lot of interaction.

Data Structures and Forms of Presentation

As explained in the topic [Classifying Lists \[Seite 397\]](#), hierarchical-sequential lists usually form the cross-section of a multidimensional data cube, where the dimensions create the hierarchical part of the data structure and the data creates the sequential part of it. You program these mixed lists with ABAP/4 statements for group changes and the like.

In the topic Lists - Colors and Lines you can find guidelines for implementing hierarchical-sequential lists. But keep in mind that there is an alternative solution to this presentation problem: You may also split the list into several simple lists, one for each group, which follow one below the other or are distributed across several screens (again, a tab control would come in handy).

Hierarchical Lists and Trees: Hierarchical Data Structures

Often, when data forms some sort of hierarchical structure, it is best to present them in a tree-like fashion. You can present such data in a format that closely resembles a list (semi-graphics), or you can present it in a more graphical form where the formal restrictions of lists need not be observed. You should choose the first case, where space requirements are tight, graphics cannot be used, the hierarchy is not too deeply nested, and if there is some sort of sequential structure imposed on the data (according to the user's point of view!). The Windows file manager is a good example of the latter case. The file hierarchy of a volume also has an imposed sequential structure, because the files and directories are sorted according to some criteria, for example, they may be sorted by alphabet or by date of last revision. On the other hand, it has a "variable" hierarchical structure that makes it inappropriate for table-like presentation forms (such an alternative presents DOS with its DIR and CD commands), as files and directories are permanently added and/or deleted. In addition, the file hierarchy is often not too deeply nested. While Windows uses graphics for displaying the file hierarchy, this hierarchy is equally well displayed in the semi-graphic list-style of the ABAP/4 list processor.

However, with hierarchies which are complicated and/or more deeply nested, you should consider using the R/3 Business Graphics. This tool also provides the graphics in a separate window which may be of advantage in many cases. However, such a presentation form also involves more complexity for the user. So, consider the advantages and disadvantages of either solution, before you use business graphics.

Nets

Nets are data structures with more complex interconnections than the hierarchical ones in trees. In their simplest form, they are just trees with loops or with only a few cross links between their branches. Here, you can try to split the nets into several trees. In more complicated cases, you have to use a graphics system like the R/3 Business Graphics which is capable of displaying nets.

Data with little Structure: Object Lists

This case is the most irregular, and it may be the most difficult when you are designing lists. For instance, the data may have a basic sequential structure, but each block of data may have a different structure so that you cannot use a table-like design. In other cases, there may be no structure at all. We can only leave this case to your imagination and urge you to try to impose some sort of structure onto the data in order to help the user find his or her way through this chaos.

Classifying Lists

In the following, we present a list classification scheme which covers the most important of the cases that we discussed. This scheme is also used in topic [Lists - Colors and Lines \[Seite 401\]](#).

Header Information

Any list can contain the following header information which describe it in more detail:

- Title
- Object information
- Action information

Object Information

The term "object information" refers to header data (as displayed on the screens) and detail data. Object information identifies the object for which the list was created. It consists of field name, field value and an optional explanatory text (check text).

Action Information

The term "action information" refers to chosen actions and results. Examples include sorting, scrolling, and totalling information.

One-Line Lists

One-line lists consist of similar lines with column titles in a header line. Certain columns can have a key function and therefore identify the lines. Examples of special lines are lines with totals or subtotals, or inserted information the user can display and/or hide. The column header can be structured hierarchically.

Multiple-Line Lists

With multiple-line lists, the line information exceeds one physical line and is thus known as a **logical line**.

The header line also extends over several physical lines.

Hierarchical-sequential Lists

Hierarchical-sequential lists are extracts from a data cube whose dimensions are determined by the number of criteria used. Each combination of criteria is further specified by a number of key figures.

Fixed and/or Extracted Criteria

The extract from the data cube is determined by a number of criteria set to particular occurrences (fixed or extracted criteria). This results in a cube which is defined by the variable dimensions; this creates the list.

Fixed criteria with their occurrences are listed as **object information** because their occurrences are constant for the entire list.

Classifying Lists

Variable Criteria

Another set of criteria is varied systematically in its occurrences (variable criteria). These characteristics are the keys.

Internal and External Criteria

- External criteria vary slowly; that is, they create groups of lines within which they are constant. They are listed with their occurrences in hierarchy information described below.
- Internal criteria vary within the groups created by the external criteria and have column headings as the key figures. They are located in the left data columns of the list body while the key figures are listed to the right.

Hierarchy Information, Groups

Key figures are often listed in addition to the external criteria. They describe the data further but are constant within the group. The description of the occurrences of external criteria with the constant key figures and their values is referred to here as "hierarchy information". It describes the "groups" in the list.

List Bodies, Key Figures

The lines of the lists within the groups consist of the internal criteria and the varied key figures. Their descriptions appear in the column headings, their occurrences and their values fill the list. The lines form a one-line or multiple-line sub-list, depending on the quantity and arrangement of the internal criteria and key figures.

The key figures are the actual data in the list.

Special Cases

Matrix Lists

Matrix lists are a special type of hierarchical-sequential list; they have both column headers and line headers. They result from classification via several criteria.

Object Lists

If the internal criteria and the key figures of a list do not determine a matrix or a similarly ordered structure, we talk about an object list.

Hierarchies

Hierarchies which have a list-like form (tables of contents, for example), can also be displayed as lists.

Hierarchy Trees

Lists can also be displayed as network or tree structures.

Diagrams

Furthermore, lists can be displayed as simple diagrams, for example, bar charts.

Cross-References

Cross-References

- [Lists - Colors and Lines \[Seite 401\]](#)
- [Lists - Hotspots \[Seite 429\]](#)
- [Lists - Symbols and Icons \[Seite 432\]](#)

Lists - Colors and Lines

[Verweise \[Seite 428\]](#)

This section discusses **Colors** and **Lines** in **ABAP/4 lists**. First, one-line lists are described. For other list categories (multiple-line lists, hierarchical-sequential lists, object lists and hierarchies), only the differences are mentioned.

The rules for the header area apply to all list categories.



This keyword is based on the list classification in [Lists - General Issues \[Seite 394\]](#).

Screen Lists vs. Printed Lists

The following guidelines only apply to screen lists.

Example Lists in LIBS

This section includes a series of example lists which show the standard use of colors and lines. You can use these examples as an aid when designing your own lists. The transaction **LIBS** allows you to display an overview of all list categories.



For technical reasons, list colors are not displayed correctly. To see correct colors, use transaction **LIBS**.

[General Information on List Design \[Seite 402\]](#)

[Color Scheme \[Seite 403\]](#)

[Header Information \[Seite 405\]](#)

[One-Line Lists \[Seite 407\]](#)

[Highlighting Columns, Lines, and Cells \[Seite 411\]](#)

[Multiple-Line Lists \[Seite 414\]](#)

[Hierarchical-Sequential Lists \[Seite 416\]](#)

[Case 1: Hierarchy Information With Few Criteria and Key Figures \[Seite 417\]](#)

[Case 2: Hierarchy Information With Many Criteria and Key Figures \[Seite 419\]](#)

[Matrix Lists \[Seite 421\]](#)

[Object Lists \[Seite 423\]](#)

[Hierarchies \[Seite 424\]](#)

[Technical Notes \[Seite 426\]](#)

Cross-Reference

General Information on List Design

General Information on List Design

Motto: Clean Up First, Then Use Colors!

Poorly structured lists cannot be improved by merely using colors. Try to come up with a clear list structure in the black and white display before you add any colors.

Arrangement

You can enhance legibility by:

- using regular alignment of objects
- grouping units of information on the screen
- separating different information units with blank lines
- inserting blank columns in lines so that the information is not too compressed..

One-Line vs. Multiple-Line Lists

Always use one-line lists instead of multiple-line lists wherever possible. With multiple-line lists, try to align subordinate lines in order to achieve a good display design.

Colors

Preliminary Considerations

The color palette was developed on the basis of the following considerations:

- Desaturated colors that harmonize with each other were chosen because they are less tiring on the eye.
- Each basic color has two shades. This way, you can structure information much better.

Using Colors

When using colors, keep the following in mind:

- To avoid a mosaic effect, use only a few colors and apply them across large areas. Eliminate any gaps between areas of different color and extend colored areas beyond the length of the field to the edges of the list.
- Restrict gaps between columns and between the list body and the group box to one character, the separator line itself.

Color Scheme

Logical vs. Physical Colors

The following table provides an overview of the (ABAP) colors available to developers, together with their descriptions. These (logical) colors are reserved for particular list components. To determine how logical colors appear on a certain platform, you use the color customizer to assign specific shades (physical colors) to the logical colors.

The table also gives the names of the physical colors (for example, yellow) as delivered with the standard setting. When coloring lists, you should refer to this setting. However, users can change each of the listed colors to suit individual preferences.

Exception: The colors COL_POSITIVE and COL_NEGATIVE should always be green or red, because they represent threshold values.

Color no.	Symbolic color name	intensified	intensified off
0	COL_BACKGROUND	Background <i>GUI-dependent</i>	Background <i>GUI-dependent</i>
1	COL_HEADING	Headers <i>grayish blue</i>	secondary headers <i>bright grayish blue</i>
2	COL_NORMAL	List body (1. stripe) <i>bright gray</i>	List body (standard coloring or every 2nd stripe) <i>almost white</i>
3	COL_TOTAL	Totals, subtotals of higher degree <i>yellow</i>	Subtotals <i>bright yellow</i>
4	COL_KEY	Key columns <i>bluish green</i>	Highlighting lines/columns <i>bright bluish green</i>
5	COL_POSITIVE	Positive threshold value <i>green</i>	Inserted lines <i>bright green</i>
6	COL_NEGATIVE	Negative threshold value <i>red</i>	Free <i>bright red</i>
7	COL_GROUP	Hierarchy header <i>violet</i>	Hierarchy information <i>bright violet</i>



The INTENSIFIED flag is now used for colors and no longer for highlighting as the name suggests. In the default setting, it converts saturated colors to desaturated. However, the user can change this setting.

Color Scheme

Explanations

- By default, INTENSIFIED is switched on.
- The saturated colors (INTENSIFIED) are darker than the normal colors (INTENSIFIED OFF).
- When defining the physical list colors, a gray or white background (COL_BACKGROUND INTENSIFIED OFF) was assumed.
- To be compatible with the earlier highlighted display, COL_BACKGROUND INTENSIFIED OFF provides blue script against the background color. With COL_BACKGROUND INTENSIFIED, the script is black.
- In the following sections, the symbolic color names are used.
- In the color customizer, the list colors are named "according to their contents".

Header Information

Title

You can output a title in the work area (if, for example, a report generates several lists).

Arrangement

- ranged left
- separated by a blank line from the details

Color

- COL_BACKGROUND INTENSIFIED (blue script on background color)

Object Information

Arrangement

- Object information consists of field name, field value and, optionally, an explanatory text (check text). Like the display on normal screens, it is a one-column, two-column or multiple-column display, depending on whether there is enough space or not. It may also contain check texts.
- As on screens, there are no leading characters between the field name and the field value.
- The object information is followed by a blank line.

Appearance

<i>Field name</i>	field value	<i>Field name</i>	field value
...
<i>Field name</i>	field value	<i>Field name</i>	field value
<i>Field name</i>	field value <i>explanatory text</i>		



<i>Account</i>	Miller	<i>Company code</i>	0001
<i>Warehouse</i>	001	<i>Storage location</i>	1
<i>Plant</i>	0001 <i>SAP Walldorf</i>		

Color

- Field name: COL_BACKGROUND INTENSIFIED (blue script on background color)
- Field value and explanatory text (check text): COL_BACKGROUND INTENSIFIED OFF (black script on background color)

Header Information

Action Information

Arrangement

- Action and parameter are displayed next to each other, separated by a colon and a blank.
- If there is enough room, action information should be output issued in two or more columns.
- The action information is followed by the list **without a blank line**.

Appearance

Action: Parameter

(that is, colon and blank between action and parameter)



Sort in ascending order: Amount

Color

- COL_BACKGROUND INTENSIFIED OFF (black script on background color)

One-Line Lists

Examples

The following figures show typical one-line lists.

List with a one-line list body, keys, a total, and threshold values

Key column:
COL_KEY intensified

Heading:
COL_HEADING intensified

Listbody (uniform):
COL_NORMAL intensified off

Material	Val.of.ord.gds.	Invoiced amount	Purch.ord.gu.
UoH-Hawa	3.381,02 DEM	3.800,23 DEM	300,000 S
Semi-fin. product	3.816,38 DEM	4.050,53 DEM	150,000 S
UoH-Hawa	134,55 DEM	134,55 DEM	50,000 S
Gear for engine	6.496,91 DEM	6.671,51 DEM	400,000 ST
Residual item	8.624,47 DEM	8.833,71 DEM	450,000 ST
Semi-fin. product	3.734,21 DEM	4.059,55 DEM	300,000 S
Residual item	836,54 DEM	1.076,89 DEM	200,000 S
Totals	27.024,080 DEM	28.717,03 DEM	1.850,00 *

Totals:
COL_TOTAL intensified

Positiv threshold value:
COL_POSITIVE intensified

Negativ threshold value:
COL_NEGATIVE intensified

One-Line Lists

The same list with stripe pattern

Key column:
COL_KEY intensified

Heading:
COL_HEADING intensified

Listbody, first stripe:
COL_NORMAL intensified

Material	Val.of.ord.gds.	Invoiced amount	Purch.ord.gu.
UvH-Hawa	3.381,02 DEM	3.800,23 DEM	300,000 S
Semi-fin. product	3.816,38 DEM	4.050,53 DEM	150,000 S
UvH-Hawa	134,55 DEM	134,55 DEM	50,000 S
Gear for engine	6.496,91 DEM	6.671,51 DEM	400,000 ST
Residual item	8.624,47 DEM	8.833,71 DEM	450,000 ST
Semi-fin. product	3.734,21 DEM	4.059,55 DEM	300,000 S
Residual item	836,54 DEM	1.076,89 DEM	200,000 S
Totals	27.024,080 DEM	28.717,03 DEM	1.850,00 *

Totals:
COL_TOTAL intensified

Positiv threshold value:
COL_POSITIVE intensified

Negativ threshold value:
COL_NEGATIVE intensified

Listbody, second stripe:
COL_NORMAL intensified off

Group Box

The list is displayed in a group box

Column Headers

Alignment

Column headers are aligned like the respective data columns. For example:

- Text information: left-justified
- Formatted number output: right-justified
- Column headers of higher levels: centered

Arrangement, Lines

- as columns separated by vertical lines (see below)
- separated from the list body by a horizontal line

Color

- COL_HEADING INTENSIFIED

Hierarchical Column Headers

If columns can be arranged in groups - possibly even across several levels - the individual levels are displayed as follows:

Lines

- The column headers are separated by vertical lines.
- The column headers of every level are separated by horizontal lines.

Color

- First header line: COL_HEADING INTENSIFIED
- Further header lines: COL_HEADING INTENSIFIED OFF

Example

Three-level Header Hierarchy

Heading 2nd level:
COL_HEADING intensified off

Heading:
COL_HEADING intensified

Heading 3rd level:
COL_HEADING intensified off

Level 1			
Level 2A		Level 2B	
Level 3AA	Level 3AB	Level 3BA	Level 3BB
146.446,00	338.102,00	838.429,00	521.944,00
57.172,00	694.602,00	381.638,00	468.302,00
196.312,00	181.326,00	104.246,00	13.455,00
180.115,00	8.887,00	181.740,00	537.259,00
649.691,00	349.193,00	674.969,00	867.428,00
921.613,00	862.447,00	418.488,00	757.836,00
998.749,00	557.116,00	373.421,00	650.685,00
515.571,00	34.993,00	759.949,00	83.654,00

Listbody (uniform):
COL_NORMAL intensified off

One-Line Lists

Key Columns

Color

- COL_KEY INTENSIFIED (uniform)

Exception: If the list contains too many colors, you should use the color of the list body (this also applies to all other list categories).

Column Display

Lines

Columns are separated by vertical lines.

Line Display (List Body)

Color

- **Standard:** COL_NORMAL INTENSIFIED OFF (uniform)
- **Exception:** If it is important to be able to follow the lines easily or if the list is very wide:
Alternating COL_NORMAL INTENSIFIED and COL_NORMAL INTENSIFIED OFF (starting with COL_NORMAL INTENSIFIED)

Separating Larger Groups of Lines

To increase transparency, you can separate different groups of lines by using horizontal lines. Such separator lines can, for example, separate different vendors or accounts.

Highlighting Columns, Lines, and Cells

Simple Highlighting

Columns - Color

- COL_KEY INTENSIFIED OFF

Lines - Color

- COL_KEY INTENSIFIED OFF
If required: COL_KEY to COL_GROUP INTENSIFIED OFF, if these are not used in the list.
 If no totals are available, also use COL_TOTAL INTENSIFIED OFF.

Totals and Subtotals

Color

- Totals: COL_TOTAL INTENSIFIED
- Subtotals: COL_TOTAL INTENSIFIED OFF
 If the subtotal and the total are far apart, you can also use COL_TOTAL INTENSIFIED.

Horizontal Lines

- To highlight individual totals and subtotals, they are separated from the remaining information by horizontal lines.
- If several similar subtotals are displayed one below another, they are not separated by a horizontal line.
- Different categories of subtotals, on the other hand, are separated by horizontal lines.

Highlighting Columns, Lines, and Cells

- If there are more than 5 levels, it is difficult to distinguish the different numbers of asterisks. You should therefore opt for the following display:
*(n) with n = level, for example, *(7) for a total of the 7th level

Highlighting Cells

Colors

- COL_POSITIVE INTENSIFIED (green, for example, positive threshold value)
- COL_NEGATIVE INTENSIFIED (red, for example, negative threshold value)
- Other colors, if available: COL_TOTAL, COL_KEY or COL_GROUP (in each case INTENSIFIED OFF)



The default settings for COL_POSITIVE and COL_NEGATIVE are predefined as shades of green and red.

Inserted Information

In some lists, you can choose a line and show its information in detail. Generally, a number of lines is "inserted" into the list to do this. This insertion can also be carried out on several levels.

Color

- COL_POSITIVE INTENSIFIED OFF
You should refrain from using multi-level colorization here because you do not know the number of levels to be included.

Flagging Inserted Information (Symbols)

Like totals, inserted information is flagged by symbols. To do this, you have to insert a column to hold this information. This allows you to distinguish inserted information on different levels.

- ">"- character
This character is repeated in the case of repeated insertion:
">>>" means insertion on the third level.
- From the fifth level, the level should be identified by a number, as suggested for totals:
>(n) mit n = level, for example. >(7) for insertion on the 7th level

Multiple-Line Lists

Multiple-Line Lists

For multiple-line lists, the rules for one-line lists apply except for the exceptions or modifications described below.

Example

Heading:

COL_HEADING intensified

Listbody, first stripe:

COL_NORMAL intensified

Order.pty.	Ship.to	p.	Name	Route				
Tr.	Uw	Sp	WADate	U	Gross weight	WU	Volumen	UE
11003		1464	FRED AG					
8545	01	01	17.04.93	X	286,000	KG	656,000	L
11004		3816	SMS & Sons Inc.					
2767	01	01	17.04.93	X	521,000	KG	111,000	L
11005		1801	Business Process Technology					
2635	01	01	17.04.93	X	3.248,000	KG	379,000	L
11006		6749	International Machines					2
9294	01	01	18.04.93	X	2.092,000	KG	706,000	L
11007		9986	Honkong Services					2
4360	01	01	18.04.93	X	2.578,000	KG	128,000	L
11008		7599	Miller United					2
5326	01	01	18.04.93	X	3.357,000	KG	685,000	L
11009		1782	Deutsche Lederhosen AG					2
2910	01	01	18.04.93	X	2.188,000	KG	575,000	L
11010		695	Allgemeine & Co KG					1
4395	01	01	19.04.93	X	3.752,000	KG	194,000	L

Listbody, second stripe:

COL_NORMAL intensified off

Column Headers

Lines

The header lines are separated as a whole by a horizontal line below the last physical header line of the list body.

If it is possible to align the table columns, you can define vertical lines between the column headers.

Column Display

Column Spacing, Lines

The columns of multiple-line lists are usually separated from each other by at least two blanks.

If it is possible to align the columns, you can also define vertical lines between the columns.

Line Display

Color

Stripe pattern from COL_NORMAL INTENSIFIED alternating with COL_NORMAL INTENSIFIED OFF (starting with COL_NORMAL INTENSIFIED) from logical lines.

Grouping of Lines

Blank Lines, Lines

Individual logical lines are neither separated by blank lines nor by horizontal lines.

You can, however, separate groups of logical lines, as in one-line lists.

Highlighting Columns

In multiple-line lists, no columns are highlighted except for key columns (see exception rule for key columns).

Exception: Columns can be highlighted, if the columns are aligned with each other and if continuous highlighting makes sense with the contents.

Hierarchical-Sequential Lists

Hierarchical-Sequential Lists

Below, only the differences with one-line lists are described.

Substructure

Two cases are distinguished:

1. The hierarchy information only consist of few criteria and key figures (orientation value: 4 criteria; transparency and storage requirement are the decisive factors).
2. The hierarchy information consist of a number of criteria and key figures.

Case 1: The hierarchy information is displayed just like object information, that is, field name, field value and, if necessary, explanatory text.

Case 2: The hierarchy information is displayed as normal line information. The descriptions of the criteria and the key figures then form a separate header line which is displayed above the normal header line.

Marking the Global List Structure

Color

The hierarchy information creates the vertical global structure of the list. It is therefore highlighted in color.

Lines

Groups are separated from each other by horizontal lines.

Recommendation

If you know beforehand that particular criteria do not vary, you should extract them from the list and place them in the header information or the hierarchy information. This makes the list more compact and the display clearer.

Case 1: Hierarchy Information With Few Criteria and Key Figures

Case 1: Hierarchy Information With Few Criteria and Key Figures

Example

Heading:
COL_HEADING intensified

Listbody (uniform):
COL_NORMAL intensified off

Mat.document	Item	Material	Plan	Batch	Q
Number	200	TB-Number	0000000001		
Type	902	Storage location	WE-ZONE		
49000000	0001	TE-MANU91	0001	0000000001	
Number	200	TB-Number	0000000002		
Type	902	Storage location	WE-ZONE		
49000003	0001	NUM-1	0001	0000000001	
49000011	0002	NUM-2	0001	0000000002	
Number	200	TB-Number	0000000003		
Type	902	Storage location	WE-ZONE		
49000016	0001	TE-MANU-92	0001	0000000001	
Number	200	TB-Number	0000000004		
Type	902	Storage location	WE-ZONE		
49000047	0001	KITKAT	0001	0000000001	
49000053	0002	KITKAT	0001	0000000002	
49000060	0003	KITKAT	0001	0000000003	
49000071	0004	KITKAT	0001	0000000004	

Hierarchy (group) heading
(field name):
COL_GROUP intensified

Hierarchy (group) data
(field value):
COL_GROUP intensified off

Column Headers

Column headers are not separated by vertical lines.

Hierarchy Information

The hierarchy information is placed before the actual data, with the same arrangement as on screens:

Appearance

List body n-1

Field name Field value Field name Field value

Case 1: Hierarchy Information With Few Criteria and Key Figures

```

...
Field name      Field value          Field name      Field value
...
List body n
...
-----

```



Number	200	TB-Number	0000000003
Type	902	Storage location	WE-ZONE
49000016	0001 TE-MANU-92	0001	0000000001

Color

- Field name of hierarchy information: COL_GROUP INTENSIFIED
- Field values and explanatory texts, if available: COL_GROUP INTENSIFIED OFF stored

List Body

Color

- One-line: COL_NORMAL INTENSIFIED OFF (always!)
- Multiple-line: as multiple-line lists (stripe pattern from logical lines)

Columns

Lines

Columns in a one-line list body are not separated by vertical lines.

End of the Hierarchy Group

Lines

Each group of lines introduced by hierarchy information is separated from the next group by a horizontal line.

Case 2: Hierarchy Information With Many Criteria and Key Figures

Case 2: Hierarchy Information With Many Criteria and Key Figures

Example

Heading:
COL_HEADING intensified

Hierarchy (group) heading:
COL_GROUP intensified

No.	TB-Number	BWA	P	T	Typ	Stor.loc.	Mat.doc.	Requmt.	TB-Date
		Item	Material				Plan	Batch	Q
200	00000000001	501	E	902	WE-ZONE	49000000			13.09.93
		0001	TE-MANU91			0001	00000000001		
200	00000000002	501	E	902	WE-ZONE	490000003			13.09.93
		0001	NUM-1			0001	00000000001		
		0002	NUM-2			0001	00000000002		
200	00000000003	501	E	902	WE-ZONE	49000016			13.09.93
		0001	TE-MANU-92			0001	00000000001		
200	00000000004	501	E	902	WE-ZONE	49000047			14.09.93
		0001	KITKAT			0001	00000000001		
		0002	KITKAT			0001	00000000002		
		0003	KITKAT			0001	00000000003		
		0004	KITKAT			0001	00000000004		

Hierarchy (group) data:
COL_GROUP intensified off

List body (uniform):
COL_NORMAL intensified off

Headers of the Hierarchy Information

Arrangement

The hierarchy information headers are placed before the header line and structured like the column headers, but with a separate column structure.

Lines

There is no separating blank line or horizontal line between the headers of the hierarchy information and the column headers.

Color

COL_GROUP INTENSIFIED

Case 2: Hierarchy Information With Many Criteria and Key Figures**Hierarchy Information****Arrangement**

The hierarchy information is placed before the actual data.

Lines

There is no separating blank line or horizontal line between the hierarchy information and the actual body of the list.

Color

COL_GROUP INTENSIFIED OFF

Matrix Lists

Example

Heading:
COL_HEADING intensified

Heading 2nd level:
COL_HEADING intensified off

Divisions				
Region	Video	Audio	Promotion	Training
France Paris	1.464,00	3.381,00	8.383,00	5.219,00
France Sud	572,00	6.945,00	3.816,00	4.683,00
Portugal	1.963,00	1.813,00	1.042,00	135,00
USA East	1.801,00	89,00	1.817,00	5.372,00
USA West	6.496,00	3.492,00	6.749,00	8.673,00
USA Central	9.215,00	8.624,00	4.184,00	7.578,00
USA South	9.986,00	5.571,00	3.734,00	6.506,00
Totals	31.497,00	29.915,00	29.725,00	38.166,00

Heading 2nd level:
COL_HEADING intensified off

Totals:
COL_TOTAL intensified

Listbody (uniform):
COL_NORMAL intensified off

Column Headers and Line Headings

Color

- Headers: COL_HEADING INTENSIFIED
- U Subordinate headers: COL_HEADING INTENSIFIED OFF

Exception: If the grouping of data results in a matrix list and the formerly key data becomes line header information, you can use COL_KEY INTENSIFIED or COL_KEY INTENSIFIED OFF (for subordinate headers) for these, in order to avoid changing colors.

List Body

Color

- COL_NORMAL INTENSIFIED OFF (uniform, standard)

Matrix Lists

Object Lists

Object lists have the same basic structure as hierarchical-sequential lists. The list body, however, can vary from list to list and within a list from group to group and can overall be very irregular.

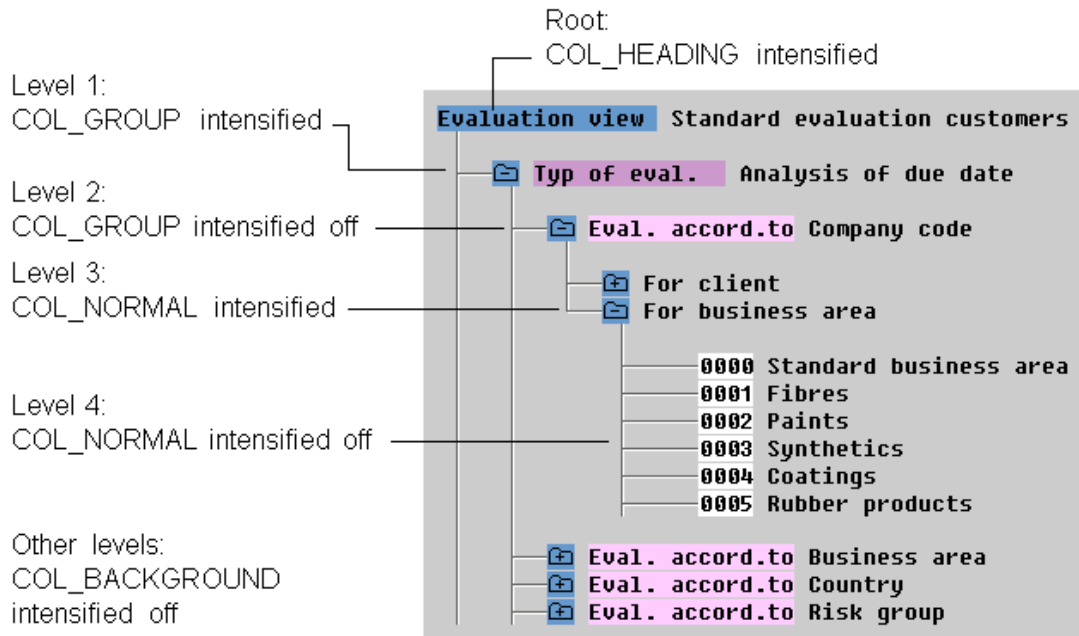
For this reason, there can be no binding rules only general recommendations for the design of object lists.

First, identify the individual information types in the list and color them according to the rules. If this proves unsatisfactory, change the basic model until you find a more aesthetically pleasing design.

Hierarchies

Hierarchies

Example



Color

Node

COL_HEADING INTENSIFIED

Level 1 and 2

COL_GROUP INTENSIFIED and COL_GROUP INTENSIFIED OFF

Level 3 and 4

COL_NORMAL INTENSIFIED and COL_NORMAL INTENSIFIED OFF

Further Levels

No storage in color (COL_BACKGROUND INTENSIFIED OFF).

Exception: If the hierarchy seems clear without color, you can refrain from using any.

For more information on the presentation of hierarchies see also [Lists - Symbols and Icons \[Seite 432\]](#).

Technical Notes

Priority of Colors

If lines as well as columns of a table are marked in color, conflicts between colors occur on the screen because color attributes cannot be combined and only one color can be displayed. The following rules determine the priority of colors in case of conflicts.

Highlightes Columns and Lines

If lines and one column are highlighted at the same time, the color of the column, usually COL_KEY INTENSIFIED OFF, has priority over the color of the lines COL_NORMAL. Therefore, the column is highlighted.

Totals and Subtotals

If lines which contain totals or subtotals are highlighted by COL_TOTAL (INTENSIFIED/INTENSIFIED OFF), this highlighting has priority over a highlighting by COL_KEY INTENSIFIED (key column) or COL_KEY INTENSIFIED OFF (highlighted data column). Therefore, totals lines are highlighted.

Highlighted Cells

The colors COL_POSITIVE INTENSIFIED and COL_NEGATIVE INTENSIFIED for highlighting cells have priority over other colors. The same applies if the colors COL_TOTAL, COL_KEY, COL_POSITIVE, COL_NEGATIVE or COL_GROUP (in each case INTENSIFIED OFF) are used for highlighting cells.

Programming Lines

Horizontal Lines

- SY-ULINE for entire lines, "-" sign for short parts in the WRITE statement.

Vertical Lines

- SY-VLINE or "|" character in the WRITE statement.

Programming Colors

Use

Colors are defined with the FORMAT command and can be used as an additional specification in the WRITE statement.

Color Name

The numbers 1-7 (for example, COLOR 3) and the symbolic names are allowed as color names (see above).

Attributes

In addition, the attributes INTENSIFIED (default) and INTENSIFIED OFF can be specified.



Please refer to the online help for FORMAT and WRITE.

Cross-References

Cross-References

- Listen - Allgemein
- [Listen - Symbole und Ikonen \[Seite 432\]](#)
- [Lists - General Issues \[Seite 394\]](#)
- [Lists - Hotspots \[Seite 429\]](#)
- Lists - Colors and Lines

Lists - Hotspots

Use and Appearance

Hotspots are areas on lists the user can click on to trigger an action.

Use

Hotspots should not be used to replace pushbuttons with no fixed position. Hotspots can not be implemented as pushbuttons because their appearance does not meet the corresponding criteria. Hotspots implemented as pushbuttons would rather confuse than support the user.

Remember: Hotspots are not pushbuttons!



For more restrictions on the use of hotspots, please refer to the *Technical Notes*!

Appearance

Hotspots are flat on the template. When the user positions the cursor on a hotspot, the cursor takes on the shape of a hand. Since this is the only way for the user to recognize a hotspot on the screen, it is recommended to indicate hotspot areas additionally with a specific color.

Area of Use

Hypertext

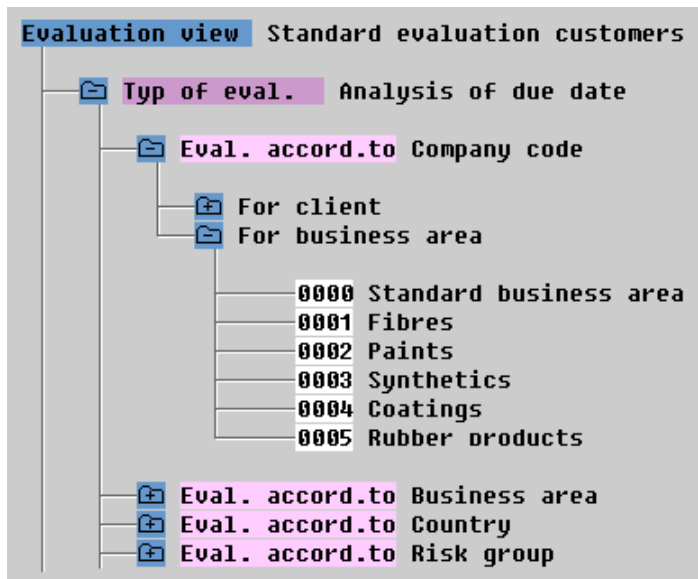
Hotspots are primarily used in hypertext links. Hotspots are to be highlighted in the text (see Technical Notes).

Hierarchies

Hotspots can be used in hierarchical lists to expand or collapse information. One example is the hierarchical graphic, another inserted information. (See [Lists - Colors and Lines \[Seite 401\]](#)).

If objects that can be expanded and collapsed should remain in an selected state for other actions, an appropriate symbol has to precede the object, for example, a folder symbol (see [Lists - Symbols and Icons \[Seite 432\]](#)), or the Expand/Collapse icons are to be used (see [Pushbuttons with Icons on Data Screens \[Seite 662\]](#)).

Lists - Hotspots



Hotspots are typically used in hierarchies (folder symbols)

Detail

In general, hotspots can be used similar to the present F2 functionality. Please note the technical restrictions (see Technical Notes).

Technical Notes

Behaviour

Hotspots are presently implemented to respond to the Mouse-Down event in the SAPGUI. The action is triggered immediately. The user cannot undo his/her action by moving the mouse and releasing the mouse button. For this reason, hotspots should not trigger actions that take a long time to execute, for example, extensive data analysis or direct manipulation of the database.

Remember: Hotspots should only be used for short-running actions which the user can undo easily and which do not change the database!

Indicating Hypertext Links

Hypertext links currently have a background color applied to them. In future, you should use a foreground color (green or blue) and underline the link.

More Information

For information on programming hotspots, please refer to the online documentation in the ABAP/4 Editor under *help hotspot*.

Cross-References

- [Lists - Colors and Lines \[Seite 401\]](#)
- [Lists - General Issues \[Seite 394\]](#)
- [Lists - Symbols and Icons \[Seite 432\]](#)

Lists - Symbols and Icons

[Verweise \[Seite 449\]](#)

As of Release 3.0 you can use small pictorial objects in lists for a wide area of application, ranging from highlighting objects up to the graphical design of lists. There are two types of pictorial elements:

- Icons
- Symbols

This section primarily deals with the use of symbols in lists, but also covers the differences between symbols and icons. You learn how to use symbols in ABAP/4 lists and/or displays which have been created using the list technology.

The SAPDings Symbol Character Set

Symbols are displayed using the SAPDings symbol character set. For an overview of the full set please see [Technical Notes \[Seite 447\]](#).

[General Remarks on Symbols and Icons in Lists \[Seite 433\]](#)

[Symbols for Displaying Hierarchies \[Seite 436\]](#)

[Symbols for Reporting \[Seite 438\]](#)

[Symbols for Office Applications \(Mail\) \[Seite 439\]](#)

[Symbols for Displaying Status \[Seite 441\]](#)

[Symbols for Indicating the Status of Objects and Values \[Seite 443\]](#)

[Symbols for Displaying Ratings \[Seite 444\]](#)

[Navigation Symbols \[Seite 446\]](#)

[Technical Notes \[Seite 447\]](#)

Cross-References



You can use transaction LIBS to display a complete list of the symbols. Here, you also find a list of printable list icons (you can display all available icons).

General Remarks on Symbols and Icons in Lists

Symbols vs. Icons - Technical and Ergonomic Differences

This section discusses the differences of the two pictorial elements from a technical and ergonomic standpoint.

Technical Differences

Icons are **colored** bitmap graphics which are available in two different sizes in the SAPGUI. You can scale icons only to a certain degree. Although you can use all icons in lists, not all icons can be printed (compare Transaction LIBS). Icons are primarily used on screens, for example, on pushbuttons. (See [Pushbuttons With No Fixed Position \(Control Element\) \[Seite 204\]](#) and [Application Toolbar \(Control Element\) \[Seite 227\]](#)).

Symbols are graphical characters which are implemented by using the SAPDings symbol character set. This is basically a character set like e.g. Courier and is available in any font. However, SAPDings consists of graphical characters. As any character set, SAPDings is available in one color and can be printed (for restrictions see below). Currently, SAPDings can only be used in lists.

Ergonomic Differences

Icons should be used in the list body in areas where the user executes an action by double-clicking or clicking (hotspot) on the icons. This applies primarily to interactive reports which are not necessarily meant to be printed. Some status icons can be printed, though. Please note that some of the icons that are arranged on top of each other, may overlap each other. This often happens with very small fonts.

Symbols should be used for lists which are to be printed. Symbols are also recommended to be used in semi-graphical lists such as hierarchy graphics: symbols can be better integrated into other pictorial elements than icons because they are freely scalable. For this reason, icons should not be used in table lists (see below).

Symbols: Areas of Application

While icons represent functions, symbols are of a more informative nature:

- Symbols can display a status:
 - describing the editing state or progress of an action
 - indicating the quality (see ratings) of an object or feature
 - indicating the availability/non-availability of objects or features
- Symbols can display a rating
 - several levels are possible (one-sided, symmetric)
- Symbols can display values in a compact and easy-to-understand way
 - for rating categories (s.a.)
 - for all kinds of units in rough categories
 - as simple scales for small object quantities or values
(more than four units cannot be recognized at a glance)

General Remarks on Symbols and Icons in Lists

- indicating errors
- Symbols can indicate objects
 - as inserted lines in lists
 - as totals (and indicate the totals level)
 - by specifying the validity of objects and values
 - by specifying the accessibility of objects and values
 - as links (in hypertext systems, glossaries, encyclopedias)
 - in exceptions: by specifying the object (this is usually covered by icons)
- Symbols direct the attention of the user towards a particular direction
 - as arrows, triangles or hands

Some of the symbols can also be used as field names and in some cases replace icons.

Consistency

Geometric signs rarely have a definite meaning. For this reason, several symbols can be used for one purpose.



Ratings can be displayed using filled or empty circles and/or circles with plus or minus signs.

This norm does not give ultimate rules and regulations. You should choose symbols that make sense in a particular context and you should be consistent with the use of symbols, no matter what symbols you are using.



Ratings scales: Circles with a plus sign go together with circles with a minus sign, however, not with filled circles or boxes.

Legends

Since symbols can be ambiguous and do not have a definite meaning, you should explain their meaning in a legend. For this purpose a centrally stored function module will be created. The user should be able to call the legend using the "i" symbol in the standard toolbar. Lists that the user prints should have the legend at the end of the printed output.

Attention, Arrangement

Since symbols are black and white characters they do not that easily catch the user's attention as icons. In addition, symbols are usually smaller, because they have to fit the normal character font.

Tabular lists often benefit from symbols, because they do not influence or even destroy the overall appearance of a list as icons. Symbols should not be distributed across a list, but should appear at fixed positions where they are easy to recognize.

General Remarks on Symbols and Icons in Lists

Colors

If you want to use color to highlight symbols, you should only color the background in accordance with the design guidelines for colors in lists. You should choose light colors because they contrast the symbol and do not disturb the overall appeal of a list.

We do not recommend the use of colors for the foreground because symbols are much too small for this.



Colors used to highlight symbols do usually not or not fully print.

Using Symbols vs. Icons in Lists

Icons trigger a function (for example, pushbutton on a screen) or indicate objects. Symbols specify the status or values of an object.

Some of the symbols and icons available overlap in their meaning. There are some icons which indicate status. The following gives recommendations for the use of the appropriate pictorial element in border line cases.

Icons

Due to their colorful and large appearance icons are eye-catching. They should be used for status displays only, if you actually want to achieve such an effect and if there are only very few status displays in the list.

If status displays appear in every line or in regular intervals, the status flags should be placed at the border when implemented as icons.



Icons can extend over more than one line. For this reason, check if they can appear at every other line to avoid overlapping.

Printing: When printing, icons are represented by graphical characters of a special character set. The restrictions for regular symbols apply. The other icons cannot be printed.

Symbols

Symbols are not as eye-catching as icons. They harmonize with the list structure. To highlight symbols, use an appropriate background color.

Symbols are scalable and adjust automatically to the font of the surrounding typeface (this also applies to printing). Symbols can be used in any list line.



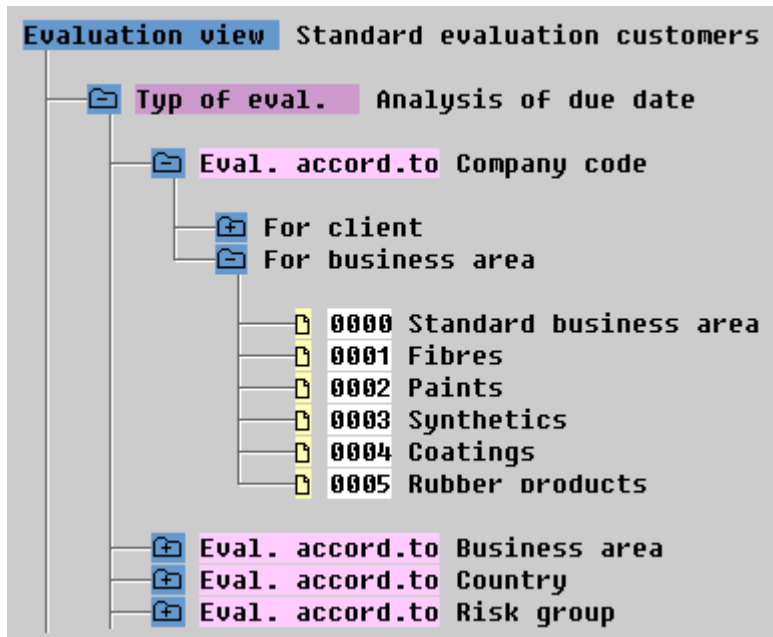
The ABAB/4 names of the symbols start with the prefix "SYM_". This prefix is omitted in the following lists!

Symbols for Displaying Hierarchies

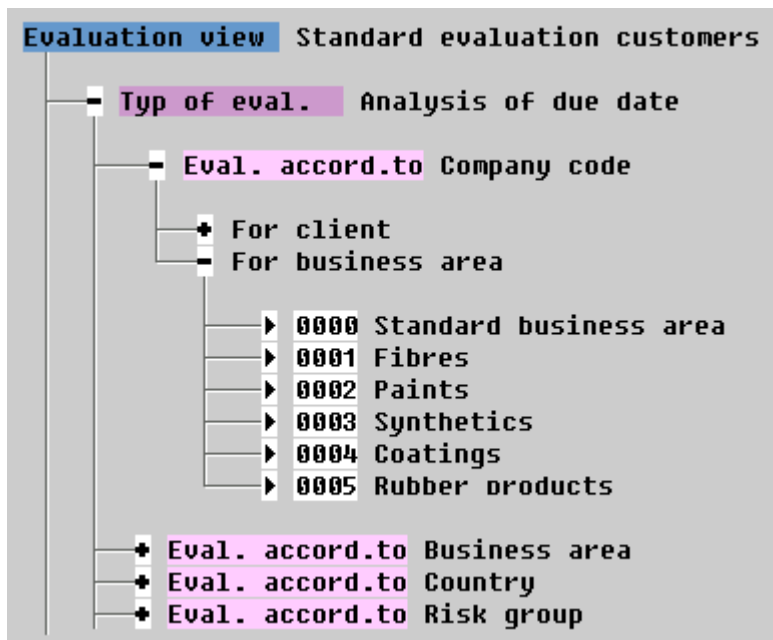
Symbols for Displaying Hierarchies

SAPDings contains all symbols necessary to display hierarchies. Symbols have the advantage that they adjust automatically to the font used.

Hierarchies in Standard Displays



Hierarchies in Compressed Displays (Reporting)



You can use this type of display when there is not much room available.













Nodes: The symbols above are binding (+, -, folder with +/-)!

Page symbol: Since hierarchies can represent many different objects, the proposals for page symbols are not binding.

Symbols for Reporting

Symbols for Reporting


























You can use the following symbols in Reporting:

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
	ASTERISK	Totals (at different levels)		DIAMOND	Subtotals
	DOCUMENT	Document		CHECKED_DOCUMENT	Checked or changed document
	DOCUMENTS	Dokuments, copy		NOTE	Note, annotation
	FOLDER	Folder		OPEN_FOLDER	Open folder
	CHECK_MARK	Status indicator		BOLD_X	Wrong or invalid value
	FILLED_CIRCLE	Mark, status indicator "positive"		CIRCLE	Mark, status indicator "negative"

Symbols for use in reporting tasks

Symbols for Office Applications (Mail)

You can use the following symbols in Office:

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
	DOCUMENT	Document		CHECKED_DOCUMENT	Checked or edited document
	DOCUMENTS	Documents, copy		NOTE	Note, annotation
	FOLDER	Folder, filing		OPEN_FOLDER	Open folder
	DOCUMENT & GLASSES	Read, display document (2 symbols)		DOCUMENT & PENCIL	Write, create document (2 symbols)
	GLASSES	Read-only		PENCIL	Read-and-Write
	CHECK_MARK	Status: read, sent, etc.		DELETE	Deletion flag, marked for deletion
	UNLOCKED	Accessible, unlocked, etc		LOCKED	Protected, locked, etc.
	RIGHT_TRIANGLE	Marking		PHONE	Phone
	PRINTER	Printer, print		FAX	Fax machine
	LEFT_HAND	Hint		RIGHT_HAND	Hint
	WORK_ITEM	Workflow item		EXECUTABLE	Executable
	FLASH	Urgent, pressing		CAUTION	Caution, warning, deadline
	DOCUMENT & FLASH	Express document (2 symbols)			

Symbols for use in office applications

Symbols for Office Applications (Mail)

**Address**

Address				
Name	[Miller]	[06229-12345]
Surname	[John]	[06229-12356]
Street	[1275, Newsman]		
City	[San Francisco, CA]	ZIP [69199]	

Symbols for Displaying Status

✓ CHECK_MARK	Edited, read, available (present), executable etc.	Empty	Unedited, unread, unavailable (absent), not executable etc.
-----------------	--	-------	---

Symbol as status display: implicitly two-state *)

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
✓	CHECK_MARK	O.k.	✗	BOLD_X	Not o.k.
●	FILLED_CIRCLE	Available	○	CIRCLE	Not available
🔒	LOCKED	Locked, protected	🔓	UNLOCKED	Unlocked, accessible

Symbol as status display: two-state (symbols in a row belong together)

● FILLED_CIRCLE	Finished, whole, full	◐ HALF_FILLED_CIRCLE	In process, partial, half	○ CIRCLE	Unprocessed, nothing, empty
--------------------	-----------------------------	-------------------------	---------------------------------	-------------	-----------------------------------

Symbol as status display: three-state

*) The binary feature can be specified with a heading (for example, "Executable")



Example 1 (two-value, two symbols): Comparison of product characteristics

Symbols for Displaying Status

	Word Processor		
Feature	FastWrite	WritePerfect	Writer
Tables	●	●	●
Spellchecker	○	○	●
Hyphenation	○	●	●
Grammar Checker	●	●	○

Example 2 (two-value, one symbol): Comparison of product characteristics (alternative)

	Word Processor		
Feature	FastWrite	WritePerfect	Writer
Tables	✓	✓	✓
Spellchecker			✓
Hyphenation		✓	✓
Grammar Checker	✓	✓	

Example 3: Status display as addition to other information











Contact	Company	Phone	Info
John Miller	Funny Food	06789-12345	✓
David Porter	Trading Post		
Barbara Turner	Used Cars Co.	0234-2345677	
Anne Windsor	Foodstore		✓



Example 1 makes a more compact overall impression than Example 2. If the characteristics refer to one (of many) columns, (Example 3), one-value status symbols can be better recognized.

Symbols for Indicating the Status of Objects and Values

Symbols for Indicating the Status of Objects and Values

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
	GLASSES	Read-only		PENCIL	Read-and-Write
	LOCKED	Protected, locked		UNLOCKED	Accessible, unlocked
	BOLD_X	Invalid, error		FILLED CIRCLE & CIRCLE	Division by zero
	DELETE	Marked for deletion		CUMULATED	Column summed up, accumulated
	SORTED_UP	Column sorted up		SORTED_DOWN	Column sorted down

Symbols showing the status of objects, numbers, data columns etc.



If a report calculates erroneous data/ data is missing/ data is invalid, then use the X symbol in the corresponding line.

Totals

Asterisks only: ***, **, ***, ****, ******* etc.

Mixed Presentation: ***, **, ***, 4*, 5*** etc.

Inserted Lines (in Lists)

▶ ▶▶ ▶▶▶ or 3▶ (several inserted layers)

Area of application: If the user double-clicks on a list line, new lines are inserted.

Symbols for Displaying Ratings

Symbols for Displaying Ratings

Ratings can be two- or multiple-level, one- or two-sided or display a ranking hierarchy:

Two-state: ⊖ negative ⊕ positive
 Multiple state (one-sided): ● ●● ●●● etc.
 Multiple state (two-sided): ... ⊖ ⊖ ⊖ ⊕ ⊕ ⊕ ...
 Ranking: ● ●● ●●● etc.



Rating user-friendliness (two-sided, 5 levels), scores and ranking order

Rating

Feature	Rating
Expectancy	○⊕⊕⊕⊕
Descriptiveness	○⊕⊕
Learnability	○⊕⊕⊕
Controllability	○
Error Robustness	○⊕
Individualization	⊖⊖⊖○

Score

Feature	Value
Expectancy	●●●●●●
Descriptiveness	●●●●●
Learnability	●●
Controllability	●●●●●
Error Robustness	●●●●
Individualization	●

Ranking Order

Feature	Rank
Expectancy	1 ●
Descriptiveness	3 ●●●
Learnability	2 ●●
Controllability	5 ●●●●●
Error Robustness	4 ●●●●
Individualization	6 ●●●●●●

Navigation Symbols

Navigation Symbols

Navigation symbols are usually displayed as icons on pushbuttons in the standard and/or application toolbars. Special applications may require additional symbols in the lists. You can use the following symbols:

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
◀	LEFT_TRIANGLE	Move back one horizontal unit	▶	RIGHT_TRIANGLE	Move forward one horizontal unit
▲	UP_TRIANGLE	Move up one vertical unit	▼	DOWN_TRIANGLE	Move down one vertical unit

Symbols for navigation functions

Technical Notes

ABAP/4

You can define the symbols of the symbol character set by using the ABAP/4 WRITE statements:

```
WRITE sym AS SYMBOL.
```

The symbolic names of the symbols are defined in the <SYMBOL> include which must be added to the report using the symbols. Transaction LIBS contains the symbolic names of the symbols in the list "List of Symbols with Proposals".



For more information on symbols in lists, please see "help write" in the ABAP/4 editor.

Special Cases

The SAPDings character set is "pseudo-equidistant". Nominally, the characters have the same width: the next character begins after the nominal width of a character. For more complex objects there is usually not enough room. For that reason, some of the characters are actually wider (double width at the maximum). These characters must be followed by a blank character to avoid overwriting the next character. The ABAP/4 WRITE statement automatically inserts a blank character. For more information on the width of the symbols please refer to the "width" column in above mentioned table.

The SAPDings character set is a 7-bit PC character set; it defines only characters in the coding area between 0 and 127.

Printing Symbols

When outputting reports with the SAPWIN/SAPLPD device, symbols can be printed in lists. With non-postscript and/or non-truetype printers and/or when printing with other devices, replacement characters are printed (see Transaction LIBS).

To print icons in lists, the SAPIcons character set is used. This character set creates replacement symbols for these icons. Above mentioned restrictions apply (that is, on some printers only replacement characters are printed). Icons for which SAPIcons has not defined replacement characters cannot be printed at all!

SAPDINGS Symbol Character Set

SAPDings 24                                

Cross-Reference

- Listen - Allgemein
- [Listen - Farben und Linien \[Seite 401\]](#)
- [Lists - General Issues \[Seite 394\]](#)
- [Lists - Hotspots \[Seite 429\]](#)
- Lists - Symbols and Icons

Functions - General Guidelines and Overview

[Zurück \[Seite 481\]](#)

This section contains general guidelines for the standard functions of the R/3 System.

[Functions - Overview \[Seite 451\]](#) provides an overview of all standard functions listed according to their use.

[Defining Functions \[Seite 456\]](#) gives general technical information on how to define functions. This information is not repeated in the following sections.

[Navigation Functions - Overview \[Seite 460\]](#) gives an overview of R/3 navigation functions and compares the basic navigation functions Back, [Cancel \[Seite 470\]](#), [Exit and/or End <application> \[Seite 492\]](#).



For an overview of all standard functions with their pull-down menus, function keys and pushbuttons as well as their syntaxes, please refer to [Overview of Standard Functions \[Seite 646\]](#).

[Functions - Overview \[Seite 451\]](#)

[Defining Functions \[Seite 456\]](#)

[Navigation Functions - Overview \[Seite 460\]](#)

Functions - Overview

In the R/3 System, standardized functions are used. Standard functions always have the same name and are called in the same way in the various tasks of the R/3 System. They are a significant element in creating a consistent user interface. You should therefore try to use these standard functions as much as possible "in accordance with these guidelines".

This section lists and describes the R/3 standard functions. It also mentions the standardized function keys and the positions of the functions in the pull-down menus.

The standard functions are classified as follows:

- Functions for editing entire objects
- Functions for comparing and storing data
- Navigation functions
- Functions for editing table structures
- Basic control functions
- Other functions

[Overview List \[Seite 452\]](#)

[Cross-References \[Seite 455\]](#)



For a general overview of the functions, their names, pull-down menus, function keys and pushbuttons as well as access characters, please refer to [Overview of Standard Functions \[Seite 646\]](#).

Overview List

[Ändern \(Funktion\) \[Seite 477\]](#)

[Anlegen \(Funktion\) \[Seite 479\]](#)

[Anzeigen \(Funktion\) \[Seite 481\]](#)

[Ausführen \(Funktion\) \[Seite 485\]](#)

[Bearbeiten, Einzelbearbeitung, Sammelbearbeitung \(Funktion\) \[Seite 489\]](#)

[Erfassen, Einzelerfassung, Sammelerfassung \(Funktion\) \[Seite 503\]](#)

[Liste \(Funktion\) \[Seite 514\]](#)

[Stornieren \(Funktion\) \[Seite 529\]](#)

[Generieren \(Funktion\) \[Seite 508\]](#)

[Merken \(Funktion\) \[Seite 519\]](#)

[Prüfen \(Funktion\) \[Seite 521\]](#)

[Sichern ohne Prüfen \(Funktion\) \[Seite 526\]](#)

[Anderes <Objekt> \(Funktion\) \[Seite 474\]](#)

[Anderes <Teilobjekt> \(Funktion\) \[Seite 476\]](#)

[Auswählen \[Seite 487\]](#)

[Erstes Bild, Letztes Bild \(Funktionen\) \[Seite 507\]](#)

[<Teilobjekt> auf/ab/links/rechts \(Funktionen\) \[Seite 531\]](#)

[Vorige Seite, Nächste Seite \[Seite 535\]](#)

[Voriges Bild, Nächstes Bild \(Funktionen\) \[Seite 536\]](#)

[Voriges <Teilobjekt>, Nächstes <Teilobjekt> \(Funktion\) \[Seite 538\]](#)

[Zurück \(Funktion\) \[Seite 544\]](#)

[Alle Mark. löschen \[Seite 473\]](#)

[Auswählen \[Seite 487\]](#)

[Ausschneiden \[Seite 486\]](#)

[Drucken \(Funktion\) \[Seite 496\]](#)

[Editor bzw. Langtext \(Funktion\) \[Seite 497\]](#)

[Widerrufen \(Funktion\) \[Seite 543\]](#)

[Rücknahme \(Funktion\) \[Seite 522\]](#)

Functions for Editing Entire Objects

- Cancel/Reverse
- Change
- Create (Function)
- [Delete or Delete <object> \[Seite 515\]](#)
- Display
- Edit, Single Processing, Collective Processing
- Enter, Single Entry, Collective Entry
- Execute
- List

Functions for Comparing and Storing Data

- [Check \[Seite 521\]](#)
- Generate
- Hold
- [Save or Post \[Seite 524\]](#)
- Save Without Check

Navigation Functions

- Back)
- [Cancel \[Seite 470\]](#)
- Choose
- [Exit or End <application> \[Seite 492\]](#)
- [First Page, Last Page \[Seite 505\]](#)
- First Screen, Last Screen
- <Object component> up/down/left/right
- [Open \[Seite 512\]](#)
- Other <object component>
- Other <object>
- [Previous Page, Next Page \[Seite 535\]](#)
- Previous Screen, Next Screen
- Previous <object component>, Next <object component>

Overview List

Functions for Editing Table Structures

Selection Functions

- Choose
- [Select All, Deselect All \[Seite 473\]](#)
- [Select Block \[Seite 494\]](#)
- [Select/Deselect \[Seite 516\]](#)

Editing Functions

- [Copy \[Seite 513\]](#)
- [Cut \[Seite 486\]](#)
- [Delete or Delete <object> \[Seite 515\]](#)
- [Insert Many \[Seite 534\]](#)
- Insert or Insert <object>
- Move
- [New Lines \[Seite 520\]](#)
- [Paste \[Seite 502\]](#)
- [Sort \[Seite 527\]](#)
- [Total \[Seite 530\]](#)

Scroll Functions

- [First Page, Last Page \[Seite 505\]](#)
- First Screen, Last Screen

Basic Control Functions

- [Help \[Seite 509\]](#)
- [Possible Entries \[Seite 499\]](#)
- [Menu Bar \[Seite 518\]](#)

Other Functions

- Editor or Long Text
- Print
- Reset
- Undo

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Access Characters \[Seite 184\]](#)
- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Defining Functions \[Seite 456\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Standard Toolbar \[Seite 307\]](#)
- [Tables - Maintenance \[Seite 383\]](#)
- [Tables - Scrolling \[Seite 376\]](#)
- [Uppercase/Lowercase \[Seite 619\]](#)

Defining Functions

Defining Functions

All functions at task level have to be integrated into the user interface (the functionality is defined in an ABAP/4 module). This keyword provides general technical information on the integration of functions into the user interface. Special notes applying to particular functions can be found in the technical notes for those functions.

[Technical Notes \[Seite 457\]](#)

[Cross-References \[Seite 459\]](#)

Technical Notes

To enable the user to initiate functions, the functions have to be integrated into the user interface using the Menu Painter. This integration is achieved by

- including the functions in pull-down menus
- assigning the functions to function keys



Permanently active functions (within primary windows), which are assigned to a standard function key, are automatically assigned to this key in the Menu Painter (for example, Help, Possible entries).

Menu Painter

In the Menu Painter, you define the following attributes: Function code, category, pull-down menu or pushbutton texts, and a mnemonic access character (see [Access Characters \[Seite 184\]](#)).

Function Code

The function code is a one to four-character letter combination which characterizes a function and which the user can enter to initiate processing. For a number of standard functions, the function code is predefined and suggested by the Menu Painter.

Function Category

The function category gives information about the flow control. The following categories are distinguished:

- "S", "N"
'S' and 'N' indicate a **system function**. System functions are not processed by the task program and executed before formal input checks are performed.
- "E"
'E' indicates a function which is to be executed in your program before the formal input check is performed.
- '' = without category
This indicates a function that is evaluated in your program and used for controlling. Formal input checks have priority over these functions. This category is assumed for all functions not especially marked.

For more information, please refer to the Menu Painter documentation on the field *Category*.

Texts

For the pull-down menus and the pushbuttons, you have to define texts which characterize the function. For a number of functions, menu texts and pushbutton texts as well as their abbreviations have been defined:

- The text in the menu is defined in the Menu Painter in the column of the same name and has a maximum of 20 characters.
- The text on the function key is also defined in the Menu Painter in the column of the same name and also has a maximum length of 20 characters (and a minimum length of 5). An "F

Technical Notes

<function key number> =" is prefixed automatically during display of the function key setting in order to achieve a consistent layout under different operating systems. This also supports users who prefer to use the keyboard (in future, this can be set by the user).

- The text is used in the [Function Key Menu \[Seite 278\]](#) and as pushbutton text.
- The pushbutton text is also used in the function key menu. The text is also defined in the Menu Painter and also has a maximum length of 20 characters. It is displayed dynamically.
- The abbreviation is to be used as part of the function name as "Text in the menu" and/or "Text on the pushbutton".

Access Character

An access character is to be assigned to the function to access the function with the fast path. The Menu Painter makes suggestions for standard functions; access characters can also be determined for your own functions. If you define an access character yourself, make sure that it is unique within the pull-down menu in question.

Pull-Down Menu Option

The function has to be assigned to the pull-down menu in which it is to be included.

Entry in the Application Toolbar

To display a function key as a pushbutton in the application toolbar, you have to define in the Menu Painter which pushbutton is to execute this function.

Icons on Pushbuttons

For a number of functions you can use predefined icons. For more information on icons on pushbuttons please refer [Pushbuttons With No Fixed Position \(Control Element\) \[Seite 204\]](#) and the individual functions.

Processing The Function Selection

To process a function selected by the user, a field of the category 'OK' has to be defined on the screen in the field list (the field name "OK_CODE" or "OK code" is generally used). After the function has been chosen, this field is set by the system to the respective function code at runtime.

Defining The Functionality

The contents of the functions are defined in the module pool in a module which is processed at the event PROCESS AFTER INPUT.

Further Information

For more information, please refer to the Menu Painter documentation in the menu *Help* → *Extended Help*.

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Access Characters \[Seite 184\]](#)
- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Formulations \[Seite 601\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Function Key Menu \[Seite 278\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Standard Toolbar \[Seite 307\]](#)
- [Uppercase/Lowercase \[Seite 619\]](#)

Navigation Functions - Overview

This section contains a graphical representation and description of R/3 standard navigation functions with regard to frequently used navigation options.

[Overview of Navigation Options \[Seite 461\]](#)

[Comparison of Exit, Back and Cancel \[Seite 464\]](#)

[Cross-References \[Seite 466\]](#)

Overview of Navigation Options

[Zurück \[Seite 481\]](#)

[Erstes Bild \[Seite 523\]](#)

[Voriges Bild \[Seite 536\]](#)

[Letztes Bild \[Seite 523\]](#)

[Voriges <Teilobjekt> \[Seite 515\]](#)

[Nächstes <Teilobjekt> \[Seite 515\]](#)

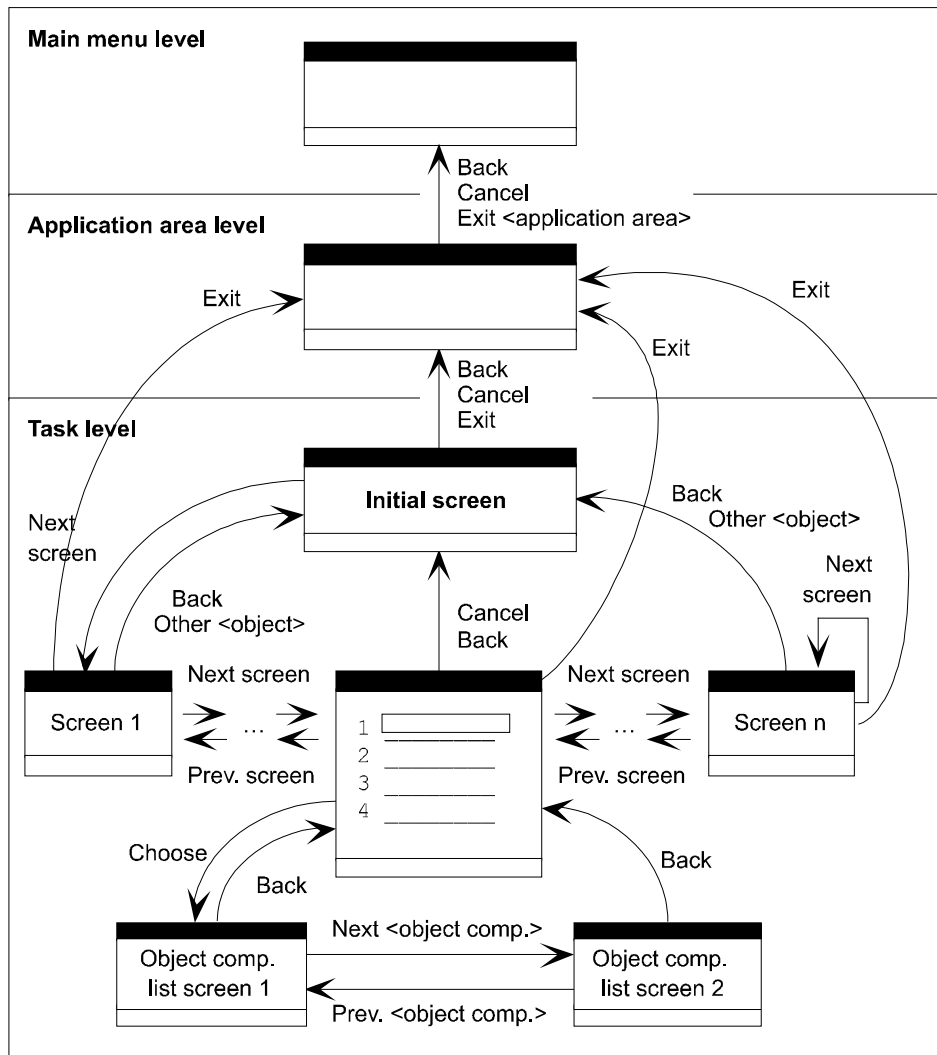
[Anderes <Objekt> \[Seite 474\]](#)

[Zurück \[Seite 481\]](#)

[Erste Seite \[Seite 505\]](#)

The R/3 System offers a large number of different navigation options as illustrated in the following graphic. For more information on navigation options please refer to [Navigation Between Screens \[Seite 106\]](#).

Overview of Navigation Options



Overview of important navigation options

Explanations: The rectangles represent screens which are allocated to the various levels of the SAP System. The arrows illustrate navigation options from one screen to another; navigation is initiated on the initial screen by the functions next to the arrow.



To maintain transparency, only commonly used options have been included in above figure. Additional options are often possible as well. In actual tasks, the options not shown can be even more useful than those shown in the graphic. You should therefore refer to the corresponding function names for options that are allowed for the respective function.

Classification of Navigation Functions

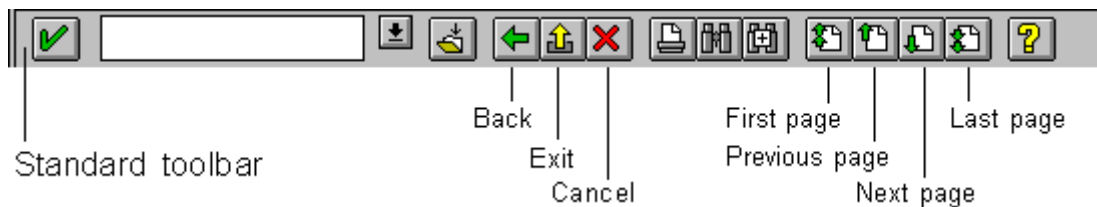
- **Elementary navigation functions:** The most important functions for return options are [Back](#), [Cancel](#) [Seite 470] and [Exit or End <application>](#) [Seite 492] and [Continue](#) [Seite 540] for forward navigation.
- **Navigation in screen sequences:** First Screen, Last Screen and [Previous Screen, Next Screen](#) [Seite 536] and/or Previous <object component>, Next <object component>.
- **Navigation in hierarchies:** [<Object component> up/down/left/right](#) [Seite 531]
- **Navigation to objects/object components:** [Other <object>](#) [Seite 474], Other <object component>, [Choose](#) [Seite 487] and [Open](#) [Seite 512].



Detailed information on these functions is provided in the topics with the respective functions names.

Navigation functions in the standard toolbar

Back, [Cancel](#) [Seite 470], [Exit or End <application>](#) [Seite 492], [First Page, Last Page](#) [Seite 505] and [Previous Page, Next Page](#) [Seite 535] are implemented as pushbuttons in the standard toolbar (see illustration) and can be initiated with the mouse.



Navigation functions in the standard toolbar

Comparison of Exit, Back and Cancel

Comparison of *Exit*, *Back* and *Cancel*

[Zurück \[Seite 481\]](#)

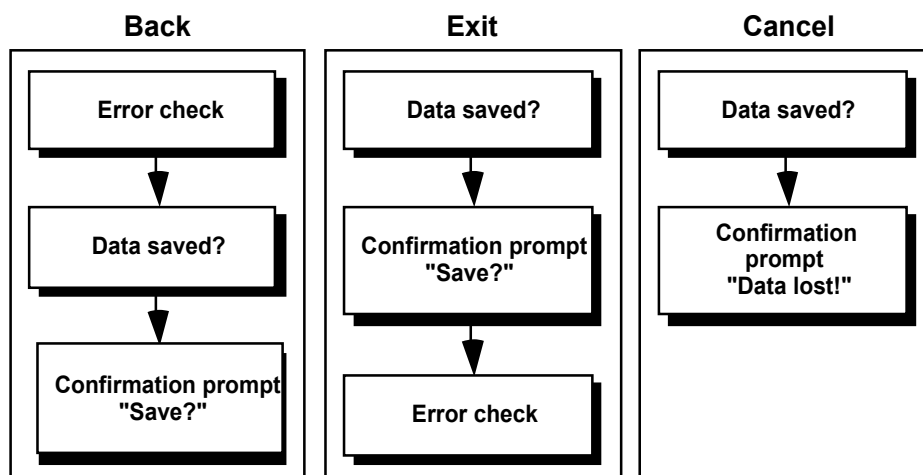
[Zurück \[Seite 481\]](#)

[Abfragen in Dialogfenstern \[Seite 50\]](#)

To clarify the differences between the three basic navigation functions Back, [Exit or End <application> \[Seite 492\]](#) and [Cancel \[Seite 470\]](#), the three functions are compared in the following.

Graphical Comparison

The following graphics show the technical procedure for the three navigation functions; in particular, they show when error checks and confirmation prompts occur.



A comparison of the three navigation functions reveals that with Back and [Exit or End <application> \[Seite 492\]](#) the error check is performed at different events: With Back, the check is performed at the beginning of the processing, whereas with Exit or End <application> it is performed at the end. With Cancel, by contrast, no check is performed.

Table Comparison

Criterion	Back	Exit	Cancel
Effect	exits object	exits task	terminates processing component
Target	returns to previous level	Screen calling the task	if possible a close screen
Save option*)	yes	yes	no
Checks	yes	yes	no
Save option	yes	yes	no

Comparison of Exit, Back and Cancel

Checks	yes	yes	no
Pull-down menu	Goto	<Object>	Edit
Function key	F3	F15	F12

*) with dialog box

Dialog Box

Criterion	Back	Exit	Cancel
Title (Example)	Back	Exit G/L document	Cancel G/L document
Diagnosis text (Example)	---	---	Unsaved data will be lost.
Text (Example)	Do you want to post the held document?	Do you want to post the held document?	Unsaved data will be lost. --- blank line --- Do you want to cancel the editing?
Pushbutton (Default)	<u>Yes</u> , No, Cancel	<u>Yes</u> , No, Cancel	Yes, <u>No</u>
Function module	POPUP_TO_CONFIRM_STEP, POPUP_TO_CONFIRM_WITH_VALUE	POPUP_TO_CONFIRM_STEP, POPUP_TO_CONFIRM_WITH_VALUE	POP_UP_TO_CONFIRM_LOSS_OF_DATA



For screen examples please refer to Prompts in Dialog Boxes.

Cross-References

Cross-References

- [Back \(Function\) \[Seite 544\]](#)
- [Cancel \(Function\) \[Seite 470\]](#)
- [Continue \(Function\) \[Seite 540\]](#)
- [Choose \(Function\) \[Seite 487\]](#)
- [Defining Functions \[Seite 456\]](#)
- [Exit or End <application> \(Function\) \[Seite 492\]](#)
- [First Page, Last Page \(Function\) \[Seite 505\]](#)
- [First Screen, Last Screen \(Function\) \[Seite 507\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Levels in the SAP System \[Seite 72\]](#)
- [Navigation Between Screens \[Seite 106\]](#)
- [<Object Component> up/down/left/right \(Functions\) \[Seite 531\]](#)
- [Other <object> \(Function\) \[Seite 474\]](#)
- [Other <object component> \(Function\) \[Seite 476\]](#)
- [Previous Page, Next Page \(Functions\) \[Seite 535\]](#)
- [Previous Screen, Next Screen \(Functions\) \[Seite 536\]](#)
- [Previous <object component>, Next <object component> \(Functions\) \[Seite 538\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)

Functions - List

This section describes the standard functions of the R/3 System. They are listed in alphabetical order.

[Overview \[Seite 469\]](#)
[Back \(Function\) \[Seite 544\]](#)
[Cancel \(Function\) \[Seite 470\]](#)
[Cancel/Reverse \(Function\) \[Seite 529\]](#)
[Change \(Function\) \[Seite 477\]](#)
[Check \(Function\) \[Seite 521\]](#)
[Choose \(Function\) \[Seite 487\]](#)
[Close \(Function\) \[Seite 523\]](#)
[Continue \(Function\) \[Seite 540\]](#)
[Copy \(Function\) \[Seite 513\]](#)
[Create \(Function\) \[Seite 479\]](#)
[Cut \(Function\) \[Seite 486\]](#)
[Delete or Delete <object> \(Function\) \[Seite 515\]](#)
[Display \(Function\) \[Seite 481\]](#)
[Display <-> Change \(Function\) \[Seite 483\]](#)
[Edit, Single Processing, Collective Processing \(Functions\) \[Seite 489\]](#)
[Editor or Long Text \(Function\) \[Seite 497\]](#)
[Enter, Single Entry, Collective Entry \(Functions\) \[Seite 503\]](#)
[Execute \(Function\) \[Seite 485\]](#)
[Exit or End <application> \(Function\) \[Seite 492\]](#)
[First Page, Last Page \(Function\) \[Seite 505\]](#)
[First Screen, Last Screen \(Function\) \[Seite 507\]](#)
[Generate \(Function\) \[Seite 508\]](#)
[Help \(Function\) \[Seite 509\]](#)
[Hold \(Function\) \[Seite 519\]](#)
[Insert Many \(Function\) \[Seite 534\]](#)
[Insert or Insert <object> \(Function\) \[Seite 498\]](#)
[List \(Function\) \[Seite 514\]](#)
[Menu Bar \(Function\) \[Seite 518\]](#)
[Move \(Function\) \[Seite 532\]](#)

Functions - List

[New Lines \(Function\) \[Seite 520\]](#)

<Object Component> up/down/left/right (Functions)

[Open \(Function\) \[Seite 512\]](#)

[Other <object> \(Function\) \[Seite 474\]](#)

[Other <object component> \(Function\) \[Seite 476\]](#)

[Paste \(Function\) \[Seite 502\]](#)

[Possible Entries \(Function\) \[Seite 499\]](#)

[Previous <object component>, Next <object component> \(Functions\) \[Seite 538\]](#)

[Previous Page, Next Page \(Functions\) \[Seite 535\]](#)

[Previous Screen, Next Screen \(Functions\) \[Seite 536\]](#)

[Print \(Function\) \[Seite 496\]](#)

[Reset \(Function\) \[Seite 522\]](#)

[Save or Post \(Function\) \[Seite 524\]](#)

[Save Without Check \(Function\) \[Seite 526\]](#)

[Select All, Deselect All \(Functions\) \[Seite 473\]](#)

[Select Block \(Function\) \[Seite 494\]](#)

[Select/Deselect \(Function\) \[Seite 516\]](#)

[Sort \(Function\) \[Seite 527\]](#)

[Total \(Function\) \[Seite 530\]](#)

[Undo \(Function\) \[Seite 543\]](#)

Overview

This section describes the standard functions of the R/3 System. They are listed in alphabetical order.

In the R/3 System, standardized functions are used. Standard functions always have the same name and are called in the same way in the various tasks of the R/3 System. They are a significant element in creating a consistent user interface. You should therefore try to use these standard functions as much as possible "in accordance with these guidelines".

The functions are arranged by keyword, in alphabetical order. Each function is described in detail. These detailed keywords cover functional descriptions, access possibilities and technical notes.



For an overview of all standard functions with their pull-down menus, function keys and pushbuttons as well as their syntaxes, please refer to [Overview of Standard Functions \[Seite 646\]](#).

Cancel (Function)

Cancel (Function)

[Zurück \[Seite 481\]](#)

[Zurück \[Seite 481\]](#)

Cancel is a navigation function which allows the user to go to the nearest screen possible without having the system to perform any checks.

Like [Back \(Function\) \[Seite 544\]](#), *Cancel* returns to the previous level in the structure of the task. The navigation option should be **as close** to the cancelled screen as possible.

Differences to Back

The basic difference to the function [Back \(Function\) \[Seite 544\]](#), is that **no checks** are performed with *Cancel*. With *Cancel*, the user can leave a screen without having completed all required fields and corrected any input errors.

Differences to Exit

A significant difference to the function [Exit or End <application> \(Function\) \[Seite 492\]](#) is the navigation option, which can to a large extent be absolutely defined for a task. With *Cancel*, the navigation option has to be specified relative to the current screen.

There is another distinction between *Cancel* and *Exit*: if data may be lost with *Exit*, the system displays a confirmation prompt which allows the user to save the data. With *Cancel*, the system displays the confirmation prompt in a dialog box to alert the user to the critical situation. However, the user is not offered an option in the dialog box to save the data. The user can only confirm or take back the *Cancel* request.

Data Loss

All unsaved data entered between leaving the screen on which *Cancel* was executed and reaching the destination are (possibly after the display of a prompt in a dialog box) "forgotten" by the task.

If the user returns to one of the screens, unsaved data will not be there.

Repeated Use of Cancel

If the user chooses *Cancel* and navigates to another screen, *Cancel* **no longer** needs to be supported. Support of the repeated use of *Cancel* is not required.

Cancelling a Pull-Down Menu

If a pull-down menu is opened, the function *Cancel* disappears and the cursor is positioned in the work area. In the case of a multi-level pull-down menu, however, only the last level disappears (see Technical Notes).

Cancelling a Dialog Box

If a dialog box is cancelled, it disappears from the user interface. Any data entered there is lost; the system does not display a confirmation prompt.



Cancelling a Primary Window

If cancelling a primary window causes a loss of data, the user is to be informed in a dialog box about the critical situation. If the user still wants to execute *Cancel* without saving data, the system should navigate to the nearest screen; the user is not required to **correct errors** or **complete required fields**. All in all, the following navigation options are possible:

- normally the screen to which the system navigates when the user chooses the Back (Function) function
- to the nearest screen where data cannot be lost
- in a fixed screen sequence, to the previous screen in the sequence (Back (Function), by contrast, would return to the previous level in the hierarchy - the initial screen, for example).

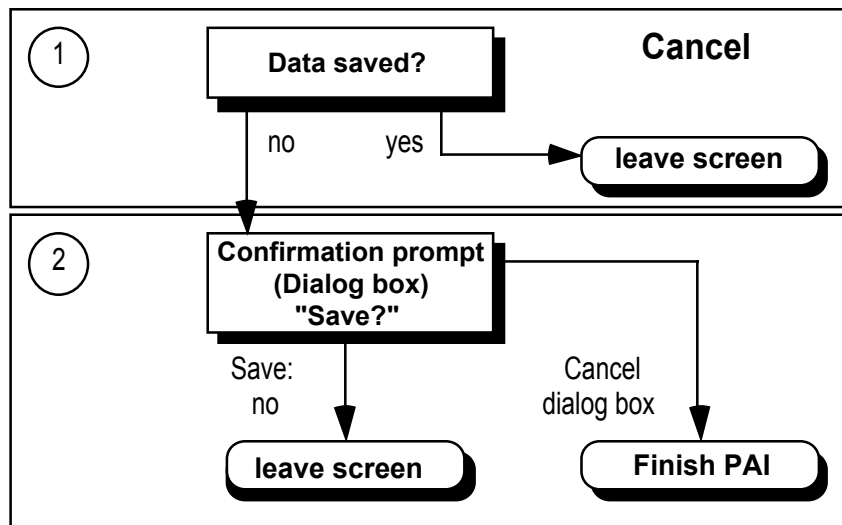
Access

The function should be always active (unless not possible for technical reasons)

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icons
F12 (F12 must not be assigned to any other function)	Last option in the menu Edit [Seite 565]	Standard toolbar (position is determined automatically)	A for <i>Cancel</i>	 (Standard toolbar)  (Dialog box)

Technical Notes

To have **no checks** performed on incorrect data, you have to provide the function *Cancel* with a function code, which is specified in the Menu Painter as type "E". In addition, first define a MODULE x AT EXIT-COMMAND in the flow logic of the screen at the event PAI. This module will be processed before all checks of the DYNP are performed. This way, the user has control and can execute *Cancel*, although data might be incorrect and other PAI modules no longer have to be processed.

Cancel (Function)

Cancelling a **pull-down menu** is automatically provided by the system and therefore does not have to be implemented by the task developer.

Cross-References

- [Back \(Function\)](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Exit or End <application> \(Function\) \[Seite 492\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Select All, Deselect All (Functions)



[Verschieben \[Seite 486\]](#)

Select all and *Deselect all* are functions at task level the user can choose select and/or deselect all the objects of a table/list. With *Select all*, the user can **select all the objects** of a table/list at one go. A function can then be performed on the selected objects, executed either simultaneously on all objects or sequentially on the individual objects, if required. For example, [Copy \(Function\) \[Seite 513\]](#), [Move \(Function\)](#) or [Delete or Delete <object> \(Function\) \[Seite 515\]](#).

With the function *Deselect all*, **all objects** are **deselected** again.

In contrast to the function [Select/Deselect \(Function\) \[Seite 516\]](#), these two functions do not consider the previous select state of the individual objects.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icons
The function key can be defined freely	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined function keys	S for <u>S</u>elect all and D for <u>D</u>eselect all	 (Select All)  (Deselect All)

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Select Block \(Function\) \[Seite 494\]](#)
- [Select/Deselect \(Function\) \[Seite 516\]](#)
- [Selection Methods - Overview \[Seite 95\]](#)

Other <object> (Function)

Other <object> (Function)

The function can display a dialog box in which the parameters of the new object can be entered. In this case, the next screen is the "first data screen" of the task. If previously entered data is not saved when *Other <object>* is chosen, and if a confirmation prompt is issued in a dialog box, then the dialog box with the confirmation prompt is displayed followed by the dialog box with the parameters.

As an exception, the function may call the initial screen of the current application (for example, if too many fields are determined on the initial screen).

Alternatively, a list/table of "entire objects" which the user had initially generated from the initial screen when calling the task can take over the function of the initial screen.



In R/3, *Other <object>* would not take you to the initial screen to the function specification of a new order but rather to the list of purchase orders generated at the beginning.

The function includes checks if data is to be saved. If an error has occurred on a screen, an error message is displayed and the user remains on that screen.



With Release 4.0A, the meaning of this function has changed. Consider these changes for your new developments.

Access

The function is to be always active (unless not always possible for technical reasons) except for on the initial screen.

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F17 *; cannot be active without corresponding key.	Menu <Object> [Seite 574]	Application toolbar: in the block of the navigation keys	O for <i>Other <object></i>	

*) F17 can, however, be used for other functions when *Other <object>* is not used in the task.

Technical Notes

You have to provide the function *Other <object>* with a function code specified as of category "E" in the Menu Painter. In addition, first define a MODULE x AT EXIT-COMMAND in the flow logic of the screen at the event PAI. This module is run through before all checks of DYNP.

If data has been changed, a confirmation prompt is to be issued. Checks should occur only when the users replies "yes" to the question "*Do you want to save the entered data?*". An identification is set for Save and the other modules of PAI are run through.

Other <object> (Function)

If data was incorrect, the user is lead to an error dialog. At the end, he can either choose the function *Other <object>* again - in which case the confirmation prompt appears once more - or he can choose the function [Continue \[Seite 540\]](#). If data is error-free, then - provided the identification *Save* is set - the data is saved at the end by PAI and the system usually navigates to the initial screen.

If the confirmation prompt is cancelled, the other modules for PAI are processed.

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Navigation Between Screens \[Seite 106\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)

Other <object component> (Function)

Other <object component> (Function)

[Voriges <Teilobjekt>, Nächstes <Teilobjekt> \[Seite 515\]](#)

[Springen \[Seite 579\]](#)

Other <object component> is a function at task level the user can choose to navigate to a line of a table/list which is to be specified in a dialog box.

With the function *Other <object component>*, the user can navigate to a fullscreen display of a line to be specified in a dialog box of a list or a table. If the individual positions of the higher-level list are documents, for example, then the function is called *Other document*.

This function can be called both from the display of the list/table and from the fullscreen display of one of the lines.

If the user initially selects lines of the list/table and then tries to activate a previously not selected object with *Other <object component>*, an error message is issued ("You can only navigate to the selected object components").

If the user wants to display the fullscreen displays of the list or table entries by accessing related navigation options, the functions [Previous <object component>](#), [Next <object component> \(Functions\) \[Seite 538\]](#) can be offered.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F20 (recommended); can be active without function key	Menu Goto [Seite 580] *	Application toolbar: in the block of the navigation keys	O for <i>Other <object component></i>	---

*) If the menu *Goto* is divided up into several menus (for example, into "Header" and "Position"), the function appears in the pull-down menu of the corresponding object components.

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- Goto (Menu)
- [Navigation Between Screens \[Seite 106\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Previous <object component>, Next <object component> \(Functions\)](#)

Change (Function)

[Anderes <Objekt> \[Seite 474\]](#)

[<Objekt> \[Seite 682\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

The user can use the *Change* function to initiate a task for changing any object of a previously selected object class.

With the *Change* function, the user can change an existing data record (Change account, Change master record). If users want to change an object of a particular object class, they have to choose the *Change* function. If they want to save the data changed during this action in a database, they have to choose *Save or Post*. *Change* must **not** be used for the data update in the database.


Like [Create \(Function\) \[Seite 479\]](#), [Display \(Function\) \[Seite 481\]](#) and [Other <object> \(Function\) \[Seite 474\]](#), *Change* navigates to the initial screen of the task.

If the user only wants to change from the Display mode to the Change mode within a display transaction to edit the current object and not to proceed via the initial screen again, the [Display <-> Change \(Function\) \[Seite 483\]](#) function can be used. This function, which appears as an option in the menu [<Object> \[Seite 574\]](#), remains on the current screen and effects the mode change described.

If the user has chosen this function, then the menu option is not active on the corresponding data screens, that is to say, it cannot be executed from the pull-down menu. The user can then restart the chosen function for another object with Other <object> (Function).

If the user has chosen *Change* is chosen, a confirmation prompt is displayed when data has not been saved and may be lost. Checks are performed if the user wants to save the data (see [Exit or End <application> \(Function\) \[Seite 492\]](#)).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function is not assigned to a particular function key	Third option in the menu <Object> [Seite 574] , right after Create (Function) [Seite 479]	The function is not displayed as a pushbutton	C for <u>C</u> hange	

Cross-References

- [Create \(Function\) \[Seite 479\]](#)
- [Display \(Function\) \[Seite 481\]](#)
- [Display <-> Change \(Function\) \[Seite 483\]](#)
- [Edit, Single Processing, Collective Processing \(Functions\) \[Seite 489\]](#)
- [Enter, Single Entry, Collective Entry \(Functions\) \[Seite 503\]](#)

Change (Function)

- Other <object> (Function),
- <Object> (Menu)
- [Save or Post \(Function\) \[Seite 524\]](#)

Create (Function)

[Ändern \[Seite 544\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

Create is a function the user can choose to start a task for creating an object of a previously selected object class.

This function can be used to create a new data record (for example, Create account, Create master record). If the user wants to create an object of a particular object class, the *Create* function is to be used. If the user wants to save the data entered during this action in the database, the [Save or Post \(Function\) \[Seite 524\]](#) function is to be used. *Create* must not be used for updating data in the database.

The functionality is not split into *Create* and *Create by Copying*.

If the user wants to create a new object using an existing object as a model, that is, to duplicate the data of the existing object, the object to be copied must be specified.

For this function, a group box with the heading *Copy from* is displayed below the block for the current object on the initial screen which was opened by choosing the function *Create*. Specifying a reference object is optional.

If a reference object is specified, the data is copied automatically onto the following screens and can then be checked manually by the user. If no reference object is specified, then the new object is created without the information of a reference object.

Create with reference is a separate function the user can choose to create a new object by referring to objects of another object class.



Please refer to [Referencing Objects \[Seite 134\]](#) for more information.


Like [Display \(Function\) \[Seite 481\]](#) and [Change \(Function\) \[Seite 477\]](#), *Create* navigates to the initial screen of the task.

If the user chooses this function, the menu option is not active on the corresponding data screens, that is, the menu option is dimmed. The user can then restart the chosen function for another object with [Other <object> \(Function\) \[Seite 474\]](#)

If *Change* is chosen within a task, a confirmation prompt is displayed in case data has not been saved and may be lost. If the data is to be saved, checks are performed (see [Exit or End <application> \(Function\) \[Seite 492\]](#)).

Create (Function)

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
This function is not assigned to a particular function key	Second option in the menu <Object> [Seite 574] (after Other <object> (Function))	This function is not displayed on a pushbutton	E for <i>Cr<u>e</u>ate</i>	

Cross-References

- Create (Function)
- [Display \(Function\) \[Seite 481\]](#)
- [Edit, Single Processing, Collective Processing \(Functions\) \[Seite 489\]](#)
- [Enter, Single Entry, Collective Entry \(Functions\) \[Seite 503\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- Other <object> (Function)
- [Referencing Objects \[Seite 134\]](#)
- [Save or Post \(Function\) \[Seite 524\]](#)

Display (Function)

[Ändern \[Seite 544\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Ändern \[Seite 544\]](#)

The user chooses the *Create* function to start a task which displays objects to be specified of a previously selected object class. Of course, only existing data records can be displayed. With the function *Display*, the data cannot be edited.


Like [Create \(Function\) \[Seite 479\]](#), [Change \(Function\) \[Seite 477\]](#) and [Other <object> \(Function\) \[Seite 474\]](#), *Display* navigates to the initial screen of the task.

Accessing the display mode with this function, the user can use the *Display <-> Change* function to access the editing mode without having to change the screen. Thus, the function [Display <-> Change \(Function\) \[Seite 483\]](#), which appears as an option in the [<object> \[Seite 574\]](#) menu, continues to refer to the previously processed object.

If the user has chosen this function, then the menu option is not active on the corresponding data screens, that is, the menu option is dimmed in the pull-down menu. The user can then restart the chosen function for another object with *Other <object> (Function)*.

If *Display* is chosen within a task, a confirmation prompt is displayed if data has not been saved and may be lost. If the data is saved, checks are performed (see [Exit or End <application> \(Function\) \[Seite 492\]](#)).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function is not assigned to a function key	Usually the fourth option in the menu <Object> [Seite 574] (after <i>Change (Function)</i>)	This function is not displayed as a pushbutton.	D for <i>D</i> isplay	

Cross-References

- [Create \(Function\) \[Seite 479\]](#)
- [Display <-> Change \(Function\) \[Seite 483\]](#)
- *Change (Function)*
- [<Object> \(Menu\) \[Seite 574\]](#)
- *Other <object> (Function)*,

Display (Function)

Display <-> Change (Function)

[Ändern \[Seite 544\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Voriges <Teilobjekt>, Nächstes <Teilobjekt> \[Seite 515\]](#)

Display <-> Change is used to switch from the Display mode to the Edit mode for changing the displayed object and vice versa or to change the processing mode of a display transaction from Display to Change or vice versa. When calling this function, no screen change is carried out and the currently active object remains active on the screen; only the processing mode changes.

In contrast to [Create \(Function\) \[Seite 479\]](#), [Display \(Function\) \[Seite 481\]](#), and Other <object> (Function), *Display <-> Change* remains on the current data screen of the task. There is no need to return to the initial screen.


Function Names

- If only switching from the display mode to the change mode is useful, the function name *Display -> Change* can be used.
- If only switching from the change mode to the display mode is useful, the function name *Change -> Display* can be used.
- If you have used the function names *Display <object>* and *Change <object>* so far, you can temporarily replace them with *Display -> Change* or *Change -> Display*. In the long term, however, the function name *Display <-> Change* is to be used.

Deactivation After the Selection

If the two functions *Display -> Change* and *Change -> Display* are available and the user chooses one of them, then the chosen function is inactive on the corresponding data screens, that is, it is dimmed in the pull-down menu. The second function, however, is active.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function is not assigned to a function key.	Menu <Object> [Seite 574] (after Save or Post (Function) [Seite 524]))	The function is not displayed as a pushbutton.	---	

Technical Notes

When switching from the display mode to the change mode, the data of the object has to be read again as changes might have been made in the meantime. The necessary locks have to be set as well.

Display <-> Change (Function)

If the user has saved an object in the Change mode with *Display <-> Change*, the system automatically switches back to the display mode and the lock on the object is released.

The automatic switch is also performed, if the user chooses *Display <-> Change* for an object component within a list of object components and switches to the Change mode. If the user, subsequently, switches to another object component with [Previous <object component>](#), [Next <object component> \(Functions\) \[Seite 538\]](#), [Next <object component>](#) or [Other <object component> \(Function\) \[Seite 476\]](#), the system is the display mode for the new object component displayed.

Cross-References

- [Change \(Function\) \[Seite 477\]](#)
- [Create \(Function\) \[Seite 479\]](#)
- [Display \(Function\) \[Seite 481\]](#)
- [Other <object component> \(Function\) \[Seite 476\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Previous <object component>, Next <object component> \(Functions\)](#)
- [Save or Post \(Function\) \[Seite 524\]](#)

Execute (Function)

Execute is a function at task level the user can choose to execute programs or reports. With this function, lists can be generated, for example. However, such sub-tasks are normally to be integrated directly into the running transaction and the technical name is not to appear in the menus but the actual function name.



Not: "Execute RFIF081A" but "Country analysis".

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu <Object> [Seite 574]	Application toolbar: in the block of the self-defined function keys	X for <i>Execute</i>	

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)

Cut (Function)



Cut (Function)

Cut is a function at task level the user can choose to process table structures. It deletes a selected area, but saves it in the clipboard.

If the user wants to transfer the selected area to the clipboard without deleting it, the [Copy \(Function\) \[Seite 513\]](#) function is to be used.

The deleted data can usually be retrieved from the clipboard and inserted at the new cursor position using [Paste \(Function\) \[Seite 502\]](#).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined functions Standard Toolbar: local cut	T for <i>Cu</i> t	 (central cut)  (Standard Toolbar: local cut)

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Copy \(Function\) \[Seite 513\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Paste \(Function\) \[Seite 502\]](#)

Choose (Function)

Choose is a function at task level the user can use to navigate to the processing of a previously selected object. List or table entries can be chosen with *Choose*. A function is then immediately executed on the selected entry.

The user will generally use *Choose* in a list of objects. After selecting one or more objects (see [Selection Methods - Overview \[Seite 95\]](#)), the user executes *Choose* to navigate to a default processing of these objects.



In the border check document, the user selects one of the lines of the document and navigates with *Choose* to a "full screen" display of the line.

Specific Naming

If a better, more specific name than *Choose* can be found for the concrete situation, this is to be used (for example, not "Choose program", when this involves the execution of the program, but rather: "Execute program").



Display, Add, Transfer

Competing Function

If functions for choosing can be executed on parts of a line as well as on the entire line, the function *Choose* is to be used for the parts of a line.



In the invoice verification document, the user can call detailed information with regard to the vendor and the client by choosing a function - regardless of the cursor position - as well as display detailed information about an entire position (that is, concerning an entire line). Choosing the position, that is, the entire line, occurs with another function (here: *Copy*), which is not on the F2 key.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F2*; not active on other key	Menu Edit [Seite 565]	Application toolbar: in the block of the standard functions	H for <i>C</i> h o ose	

*) if *Choose* is not active, then F2 can also be used for other functions



The function assigned to F2 can also be initiated automatically by double-clicking on it.

Choose (Function)

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Selection Methods - Overview \[Seite 95\]](#)

Edit, Single Processing, Collective Processing (Functions)

[Erfassen \[Seite 503\]](#)

[Erfassen \[Seite 503\]](#)

[Ändern \[Seite 544\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

Edit is a function the user can choose to start a task for entering or editing an object from a previously selected object class. With this function, one or more data records can be entered and edited. This means that the user can enter data for a new object, but also change data stored in the database for new or existing objects. The emphasis is on the **changing** of data. The term to be used is *Edit*.

If the user wants to update the data entered or changed within this function in the database, the [Save or Post \(Function\) \[Seite 524\]](#) function is to be used. *Edit* must not be used for the data update in the database.

Single Processing

Single processing is a function the user can choose to call a task for processing a **single object** of a previously selected object class. This function name is to be used instead of *Edit*, if you want to stress that a single object is being processed or if a distinction is made between Single Processing and the processing of many objects in a separate task.

Collective Processing

The *Collective processing* function is used to process many objects **at one go**. This function name is to be used instead of *Edit*, if you want to stress that many objects are being processed or if a distinction is made between *Collective processing* and the processing of a single object in a separate task.

Action Name vs. Screen Name

Single Processing and *Collective Processing* are action names for an entire task. If processing is restricted to an individual screen within a task, the function the user uses to call the task is [Create \(Function\) \[Seite 479\]](#) or *Edit, Single Processing, Collective Processing (Functions)*, depending on which aspect you want to stress. The fact that this one screen is used for single or collective processing is reflected in the screen name specified in the title bar (see [Title Bar \[Seite 152\]](#)).

Other than the mentioned characteristics, *Single Processing* and *Collective Processing* are identical to the *Edit* function.

Difference to Create and Enter

In contrast to [Create \(Function\) \[Seite 479\]](#), *Edit* allows the user to create and change data records.

Edit, Single Processing, Collective Processing (Functions)

[Enter, Single Entry, Collective Entry \(Functions\) \[Seite 503\]](#) focus on the entry of new data records whereas *Edit* also allows the user to change data.

Other Functions

Other functions for entering new data records are [Create \(Function\) \[Seite 479\]](#) and [Enter, Single Entry, Collective Entry \(Functions\) \[Seite 503\]](#).

Navigation to the Initial Screen

Like [Create \(Function\) \[Seite 479\]](#), [Change \[Seite 477\]](#), [Display \[Seite 481\]](#) and [Other <object> \[Seite 474\]](#), Edit, Single Processing and Collective Processing navigate to the initial screen of the task.

Deactivation After the Selection

If the user chooses this function, the menu option is not active on the corresponding data screens, that is, the menu option is dimmed in the pull-down menu. The user can then restart the chosen function for another object with Other <object>.

Effect as During Exit

If *Edit*, *Single Processing* or *Collective Processing* are chosen within a task, a confirmation prompt is displayed if data has not been saved and may be lost. If the data is saved, checks are performed (see [Exit or End <application> \(Function\) \[Seite 492\]](#)).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function is not assigned to a function key	Second option in the menu <Object> [Seite 574] (after Other <object> (Function))	The function is not displayed as a pushbutton	E for <i>E</i> <u>dit</u>	---

Cross-References

- [Change \(Function\)](#)
- [Create \(Function\) \[Seite 479\]](#)
- [Display \(Function\) \[Seite 481\]](#)
- [Enter, Single Entry, Collective Entry \(Functions\) \[Seite 503\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Other <object> \(Function\)](#)
- [Save or Post \(Function\) \[Seite 524\]](#)
- [Title Bar \[Seite 152\]](#)

Exit or End <application> (Function)**Exit or End <application> (Function)**

Exit or *End* <application> is a navigation function the user can choose to leave the current task or application. *Exit* leaves the current task and returns to the screen which has called the task, usually the screen of the application level.



In the application level this function is referred to as *End* <application>. The user has to explicitly specify what is exited (for example, Exit document processing, Exit Screen Painter, Exit material master). The user then returns to the main menu level.

If data may be lost when leaving the task, a dialog box appears displaying a confirmation prompt.

If the task was called, *Exit* returns to the calling screen. If the task was called with "LEAVE TO TRANSACTION", it is up to you whether you want to guide the user to the last screen before the task or to another screen. It is recommended, however, to guide the user to a consistent screen in a task area (several transactions) after *Exit* (for example, to the initial screen or the screen of the application menu).


Difference to Function Code "N"

With *Exit*, PAI modules are possibly still being processed and therefore checks are performed.

Exception: On initial screens, neither a check for incorrect data is performed nor a confirmation prompt is displayed, unless the task only consists of the initial screen.

Access

The function has to be always active. **Exception:** In dialog boxes, there is no *Exit* function.

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F15 *; not active without function key	Last option in the menu <Object> [Seite 574]	Standard toolbar (position is determined automatically)	X for <i>Exit</i>	 (Standard Toolbar)

*) F15 must not be assigned to another function.

Technical Notes

You have to provide the function *Exit* with a function code which is indicated as of category "E" in the Menu Painter. In addition, first define a MODULE x AT EXIT-COMMAND in the flow logic of the screen at the event PAI. This module is processed before all checks of DYNP so that you have control before DYNP.

If data was changed, a confirmation prompt is to be displayed. Checks should only be performed when the users replies "yes" to the question "*Do you want to save the entered data?*". An identification is set for Save and the other modules of PAI are processed.

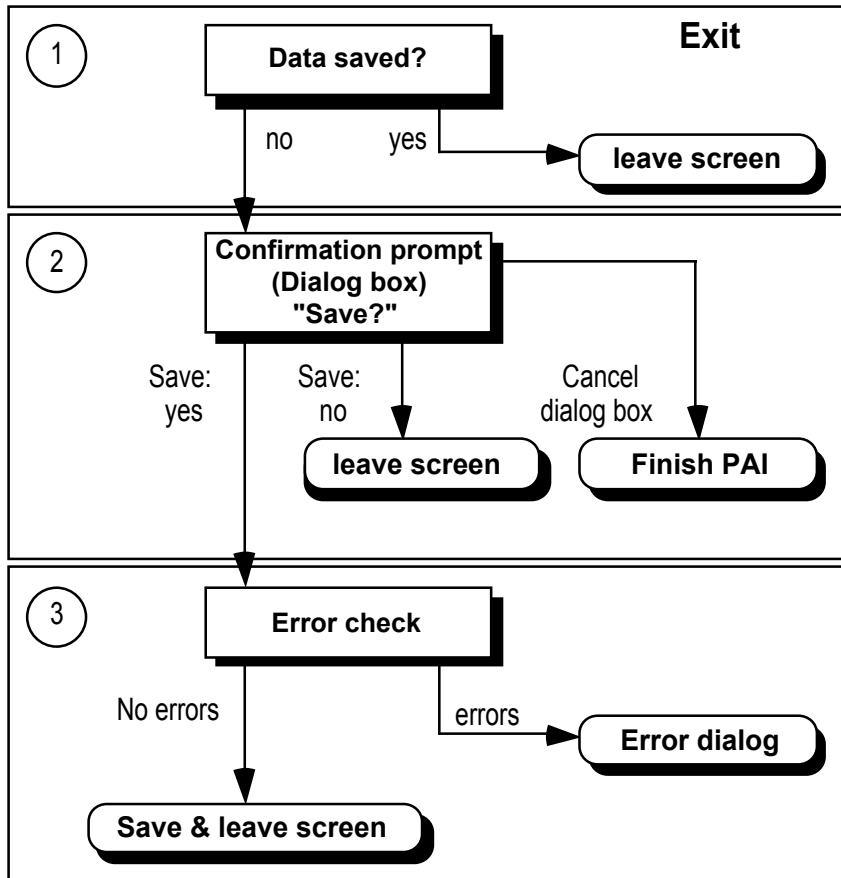
If data was incorrect, the system starts an error dialog. At the end, the user can either choose the function *Exit* again - the confirmation prompt appears again - or choose the function [Continue](#)

Exit or End <application> (Function)

(Function) [Seite 540]. If the data is free of errors, then - provided the identification Save is set - the data is saved at the end by PAI and the task is left.

If the confirmation prompt is answered with "No", the task is left without saving the data.

If the confirmation prompt is cancelled, the other modules for PAI are processed.



Cross-References

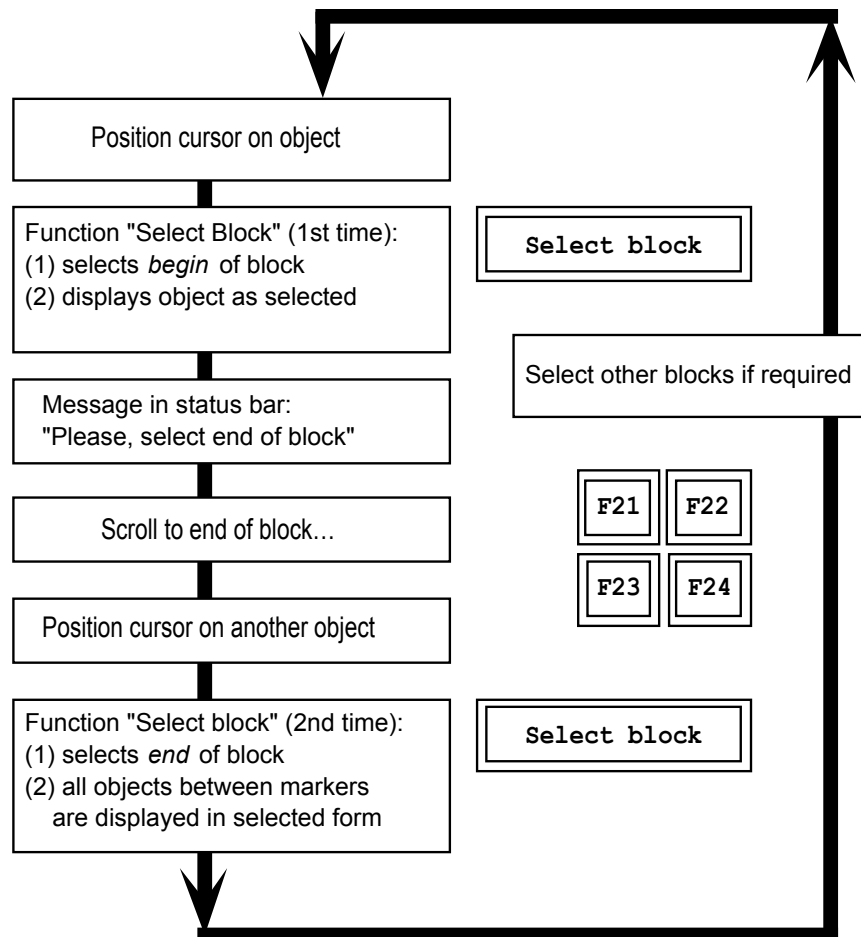
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Select Block (Function)

Select Block (Function)

[Verschieben \[Seite 486\]](#)


Select block is a function the user can choose to process tables or lists. With the function *Select block*, the user can select any block within a table/list as illustrated in the following figure.



- If two blocks overlap, the objects "selected twice" remain selected.
- Choosing any other function - except for scrolling - between the first call of the function *Select block* and the second call cancels the block selection mode and cancels the selection of the object at the beginning of the block. Already selected objects remain selected.
- Existing selections can only be reset together or individually but not in blocks. To allow resetting of blocks, the block selection function must be able to select or deselect depending on the select state of the object, as the [Select/Deselect \(Function\) \[Seite 516\]](#) function does with individual objects. In this case, however, you would not be able to select overlapping blocks.

For a selected object, the user can then initiate a function with the function keys, menu options or pushbuttons. This can, for example, be the function [Copy \(Function\) \[Seite 513\]](#), [Move \(Function\) \[Seite 532\]](#) or [Delete or Delete <object> \(Function\) \[Seite 515\]](#).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Edit [Seite 565]	Application toolbar: in the block of the self-defined function keys	B for <i>Select <u>b</u>lock</i> (default)	

Cross-References


- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Copy \(Function\) \[Seite 513\]](#)
- [Delete or Delete <object> \(Function\) \[Seite 515\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Move \(Function\)](#)
- [Select All, Deselect All \(Functions\) \[Seite 473\]](#)
- [Select/Deselect \(Function\) \[Seite 516\]](#)
- [Selection Methods - Overview \[Seite 95\]](#)

Print (Function)

Print (Function)

Print is a function the user can choose to print information from a task. With the function *Print*, the user can print a list, document, purchase order, and so on.

Access

Function Key	Pull-Down Menus	Pushbutton	Access Character	Icon
CTRL + P	Menu <Object> [Seite 574] , menu Edit [Seite 565] (Print object component)	Application toolbar: in the block of the standard functions	P for <i><u>P</u>rint</i>	

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)


Editor or Long Text (Function)

[Springen \[Seite 579\]](#)

Editor and *Long text* are functions at task level the user can choose to process texts.

The user chooses the *Editor* function to edit lines or objects (for example, editing a program), and "Long text" to enter a long text for an object. The functions can also be combined to *Editor/long text*. This function leads into an edit mode, that is, the SAP editor or the SAP word processing system or an editor-type system is generally accessed.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F9 * (recommended)	Menu Goto [Seite 580]	Application toolbar: in the block of the standard functions	E for <u>E</u> ditor, L for <u>L</u> ong text	

*) If two functions are assigned to the key, it has priority over [Select/Deselect \(Function\) \[Seite 516\]](#).

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Select/Deselect \(Function\) \[Seite 516\]](#)
- Goto (Menu)

Insert or Insert <object> (Function)

Insert or Insert <object> (Function)


This function is used for inserting single blank lines into a table or list which can be filled later. The object can, for example, be a "line" or an "position". In table structures, you can use the function name *Insert line* in accordance with the *Insert <object>* functional description.

The blank line is inserted **before** the cursor position.

If the user has not entered data before pressing ENTER, the blank line disappears again.

If several blank lines are to be inserted, the user can choose [Insert Many \(Function\) \[Seite 534\]](#) or [New Lines \(Function\) \[Seite 520\]](#). The first of the two functions inserts blank lines at any position in the table structure, the latter at the end of the table structure.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icons
The function key can be defined freely	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined functions	I for <i>Insert</i>	 (Insert Line)

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Insert Many \(Function\) \[Seite 534\]](#)
- [Tables - Maintenance \[Seite 383\]](#)
- [New Lines \(Function\) \[Seite 520\]](#)

Possible Entries (Function)

[Wertehilfe-Drucktaste \[Seite 759\]](#)

Possible entries displays the valid input values for a field.

When the user chooses the function *Possible entries*, the system displays the possible input values for a field (values, check table, matchcode), provided they were stored by the developer. The help system decides as to which possible entries are displayed. *Possible entries* is a standard function.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F4 *	The function is not assigned to a pull-down menu. **	---	---	---

*) F4 must not be assigned to any other function.

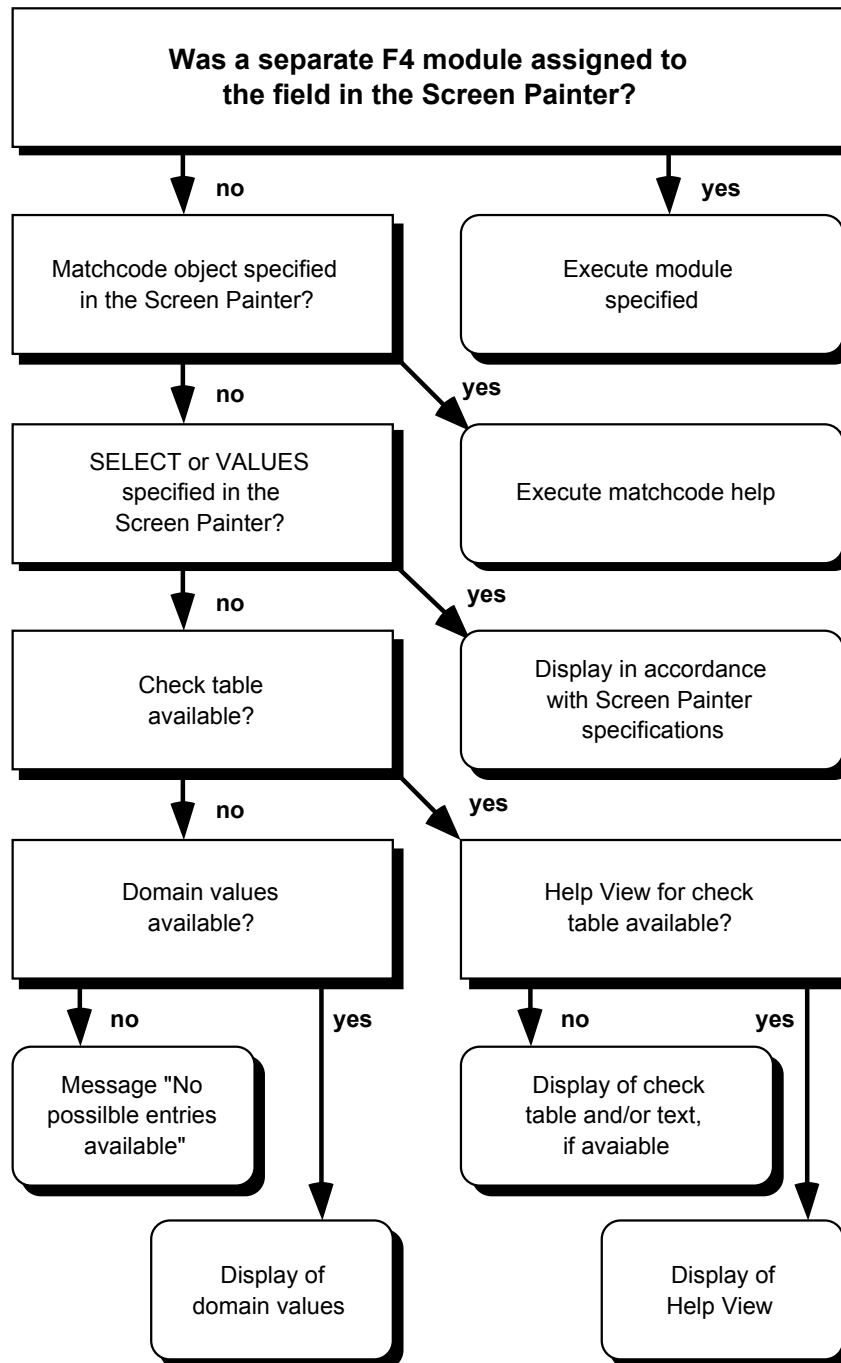
**) Further functionalities of the help system are given in the *Help* menu.



You can get help for possible entries for input/output fields with the Possible entries pushbutton. This pushbutton is permanently or not permanently displayed to the right of these fields, if the function is supported (see [Possible Entries Pushbutton \(Control Element\) \[Seite 331\]](#)).

Technical Notes

If the user calls the function *Possible entries* on a screen field ready for input, the following algorithm runs internally:

Possible Entries (Function)

Thus, the procedure after calling *Possible entries* basically depends on which specifications were made by the task developer in the Screen Painter or in the ABAP/4 Dictionary. Below, special relevant help concepts are described in more detail.

Help Views

In most cases, a help view is displayed after calling *Possible entries* which is determined with the check table specified for a field in the ABAP/4 Dictionary (The check table does not have to be

Possible Entries (Function)

used for the input validation, that is, the corresponding indicator does not have to be set in the Screen Painter.). A help view is created with the ABAP/4 Dictionary transaction SE11. The name has to be set up as follows: H_<Name of the check table>. The check table has to be specified as primary table of the help view.

Due to performance reasons, it is absolutely necessary to define a help view for each of the check tables called with F4. The key fields of the check table as well as possibly existing texts from a corresponding text table are also displayed when pressing F4 even if no help view exists, but the response time here is much longer (because a report has to be generated for the process).

Separate F4 Module

In some cases, you cannot directly control the possible entry functionality with ABAP/4 Dictionary definitions, but you have to create your own module in the task program which is executed when you call "Possible entries". In this module, data can be listed and displayed individually. The display, however, should refer to the standard display of possible entries to avoid displaying different screens to the user. To do this, you should possibly use the general function module `HELP_VALUES_GET_WITH_TABLE`. The function group SHL3 contains even further function modules which can be useful when you program your own possible entries function.

To make sure that a module is executed when you call "Possible entries", you have to specify the following in the flow logic of the Screen Painter:

```
PROCESS ON VALUE-REQUEST.
  FIELD field name MODULE module name.
```

The event `PROCESS ON VALUE-REQUEST` is always processed if the user has called "Possible entries". If the cursor is positioned on a field which is named in a following `FIELD` statement, the module specified there is executed.

Cross-References

- [Function Keys - Overview \[Seite 267\]](#)
- [Help \(Function\) \[Seite 509\]](#)
- [Help \(Menu\) \[Seite 570\]](#)
- Possible Entries Pushbutton (Control Element)



Paste (Function)

Paste (Function)

[Bearbeiten \[Seite 565\]](#)

This function is used to insert lines or areas from the clipboard directly **before** the cursor position. It is required that data was transferred to the clipboard with the [Cut \(Function\) \[Seite 486\]](#) or [Copy \(Function\) \[Seite 513\]](#) functions.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu Goto	Application toolbar: in the block of the self-defined functions Standard Toolbar: local paste	P for P aste (default)	 (central paste)  (Standard Toolbar: local paste)

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Cut \(Function\) \[Seite 486\]](#)
- [Copy \(Function\) \[Seite 513\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)

Enter, Single Entry, Collective Entry (Functions)

[Bearbeiten \[Seite 529\]](#)

[Ändern \[Seite 544\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

Enter is a function the user can choose to call a task for entering objects of a previously selected object class. The user can choose this function to enter one or more new data records. Although data can also be changed, the emphasis is on the **entering** of data for a new object.

If the user wants to update data entered within this action in the database, the function [Save or Post \(Function\) \[Seite 524\]](#) applies. *Enter* must not be used for updating data in the database.

Single Entry

Single entry is a function the user can choose to enter data of a **a single object** of a previously selected object class. This function name is to be used instead of *Enter* if you want to stress that a single object is being entered or if a distinction is made between *Single entry* and the entry of many objects in a separate task.

Collective Entry

The *Collective entry* function is used to enter many objects **at one go**. This function name is to be used instead of *Enter* if you want to stress that many objects are being entered or if a distinction is made between *Collective entry* and the entry of a single object in a separate task.

Action Name vs. Screen Name

Single Entry and *Collective Entry* are action names that apply to an entire task. If the entry is restricted to an individual screen within a task, the function the user uses to call the task is [Create \(Function\) \[Seite 479\]](#) or *Enter*, depending on what aspect you want to stress. The fact that only this screen is a single entry or collective entry screen is reflected in the screen name specified in the title bar (see [Title Bar \[Seite 152\]](#)).

Besides the mentioned characteristics, the functions *Single Entry* and *Collective Entry* are identical to the *Enter* function.

Difference to Create and Edit

Compared to [Create \(Function\) \[Seite 479\]](#), *Enter* is much more efficient for "entering" new data records, because it is designed for object classes for which data records are entered very often and in large quantities.

[Edit, Single Processing, Collective Processing \(Functions\) \[Seite 489\]](#) focus on the changing of data whereas *Enter* concentrates on entering new data records.

Other Functions

Other functions for entering new data records are [Create \(Function\) \[Seite 479\]](#), *Edit*, *Single Processing* and *Collective Processing*.

Enter, Single Entry, Collective Entry (Functions)**Navigation to the Initial Screen**

Like [Create \(Function\) \[Seite 479\]](#), [Change \(Function\) \[Seite 477\]](#), [Display \(Function\) \[Seite 481\]](#) and [Other <object> \(Function\) \[Seite 474\]](#), *Enter, Single Entry* and *Collective Entry* navigate to the initial screen of the task.

Deactivation After the Selection

If the user chooses this function, the respective menu option is not active on the corresponding data screens, that is, the menu option is dimmed in the pull-down menu. The user can then restart the chosen function for another object with *Other <object> (Function)*.

Effect as During Exit

If *Enter, Single Entry* or *Collective Entry* are chosen within a task, a confirmation prompt is displayed if data has not been saved and may be lost. If the data is saved, checks are performed (see [Exit or End <application> \(Function\) \[Seite 492\]](#)).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function is not assigned to a function key.	Second option in the menu <Object> [Seite 574] (after <i>Other <object> (Function)</i>)	The function is not displayed as a pushbutton	---	---

Cross-References

- [Display \(Function\) \[Seite 481\]](#)
- [Change \(Function\)](#)
- [Create \(Function\) \[Seite 479\]](#)
- [Edit, Single Processing, Collective Processing \(Functions\)](#)
- [Other <object> \(Function\)](#)
- [Save or Post \(Function\) \[Seite 524\]](#)
- [Title Bar \[Seite 152\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)

First Page, Last Page (Function)

[Symbolleiste \[Seite 307\]](#)



With the function *First page*, the user scrolls to the first page in a displayed list or table. Even if the user executes the function repeatedly, the system remains on the first page.

Other functions for scrolling in tables/lists are *Previous page* (F22), *Next page* (F23) and *Last page* (F24).

Last page is a function at task level the user can choose to scroll to the last page in a displayed list or table. If the last page is displayed on the screen, the function does not provoke any further scrolling action; the system does not go to the first page.

Other functions for scrolling in table structures are [Previous Page, Next Page \(Functions\) \[Seite 535\]](#).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
CTRL + PAGEUP CTRL + PAGEDOWN	The function is not assigned to a pull-down menu	Standard Toolbar [Seite 307] (position is determined automatically)	---	 (First Page)  (Last Page)

Technical Notes

The function module SCROLLING_IN_TABLE is to be used for scrolling in a table structure. Among other things, the starting index and the function code for the required scrolling function are transferred to this function module. The scrolling itself is not performed by this function module. It only determines the index of the first table line to be displayed.

Cross-References

- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Tables - Scrolling \[Seite 376\]](#)
- Standard Toolbar
- [Previous Page, Next Page \(Functions\) \[Seite 535\]](#)

First Screen, Last Screen (Function)

[Springen \[Seite 579\]](#)

[Springen \[Seite 579\]](#)



First screen is a navigation function at task level which navigates to the first screen in a fixed screen sequence.

If the system runs through a fixed screen sequence, the user can choose *Last screen* to navigate to the last screen of the screen sequence.

If there is a fixed screen sequence the system runs through, *First screen* navigates to the first screen of this screen sequence. If the user is not guided through a fixed screen sequence, the required screens can only be accessed explicitly by activating the specified options in the [Goto \[Seite 580\]](#) menu or by using the corresponding function keys.

Other scroll functions to be used in a fixed screen sequence are [Previous Screen, Next Screen \(Functions\) \[Seite 536\]](#)

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
can be defined freely	Menu Goto	Application toolbar: in the block of the self-defined function keys *	---	 (First screen)  (Last screen)

*) not F21 and/or F24!

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- Goto (Menu)
- [Navigation Between Screens \[Seite 106\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Previous Screen, Next Screen \(Functions\) \[Seite 536\]](#)


Generate (Function)

Generate (Function)

Generate is a function at task level the user can choose to generate an object, for example, a screen.

When "generating" an object (for example, a screen), the new object is automatically saved and a consistency check is performed.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
can be defined freely	Menu <Object> [Seite 574]	Application toolbar: in the block of the self-defined function keys	T for <i>Generate</i>	

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)

Help (Function)


[Hilfe \[Seite 509\]](#)

Help displays a help text for an object the user has selected with the cursor.

The function *Help* displays a help text. The help system, which was activated by choosing this function, determines the text to be displayed. *Help* is a standard function.

Access

The function is to be always active.

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F1 *	The function is not assigned to a pull-down menu **	Standard Toolbar [Seite 307] (position is determined automatically)	---	 (Standard Toolbar)

*) F1 must not be used for any other function

**) (further functionalities of the help system are given in the menu [Help \[Seite 570\]](#))

Technical Notes

When the user chooses the *Help* function, usually the help system has the control, not the task program. Often, however, you can influence the process and the displayed documentation from the task development point of view as follows:

After the user calls *Help*, the help system determines the documentation related to the current object and displays it. The documentation can be created in one of two ways:

- using transaction SE61
- using the development tool of the object to be documented (for functions, for example, in the Menu Painter).

The help system determines the help text to be displayed based on the technical information of the chosen object. This identification is always consists of the following elements for the individual objects:

For menu options Program and function code
or function keys:

For messages: Message ID and message number

For input fields: Data element

Thus, not the field names are used for input fields on the screen but the data element which is allocated to the table field specified on the screen. Consequently, the display of a help text for a

Help (Function)

field can only be used as a default, if the field has a ABAP/4 Dictionary reference (exception: see below). Generally, the data element documentation is displayed.

Since the data element documentation is in some cases insufficient, you can enter data element supplements which are then displayed in the help system together with the data element documentation. Here, you generally have to distinguish between general and screen-specific data element supplements. If general data element supplements are created, these are usually displayed together with the definition. If you also create screen-specific data element supplements, they are prioritized and displayed together with the definition instead of the general data element supplements.

Data element supplements are created during the maintenance from the F1 help. They are allocated via table THLPF where the table entry is created automatically during the maintenance. When maintaining, you have to enter: Program, screen, table field, data element supplement.

If there is no ABAP/4 Dictionary reference or the functionality of the standard help is not sufficient, you can program the help for an input field in the task program itself. In this case, it is linked via the above-mentioned event POH. If the FIELD...MODULE... statement does not contain "WITH", only the specified module is executed but not the standard help system.

Special Features for the Help Function in Reports

Linking field documentation in reports differs in some points from procedures used with transactions. Reports can contain the following fields: parameters, select options and fields in lists.

The developer does not define screens in reports, that is, there is no event POH. Currently, there is also no equivalent event available so you cannot program the help yourself. (However, you can occupy the F1 key in the Menu Painter but this way you switch off the standard help for the entire interface status.)

Report fields are linked to a table field - and therefore to a data element - as follows:

for parameters:	LIKE (for example, ZZ LIKE T001-BUKRS)
for select options:	FOR (for example, UMSATZ FOR KUMUM)
for fields in lists:	WRITE table field (where table has to be a ABAP/4 Dictionary table)

For fields in lists which refer to the ABAP/4 Dictionary in its definition with "LIKE" or "INCLUDE STRUCTURE", an automatic help link is planned.

When allocating data element supplements to objects via table THLPF, the report name is always to be entered as a program. For parameters and select options, you always have to specify screen "1000", regardless of the screen on which the parameters or the select option actually appear. For fields in report lists, you have to specify screen "120".

Display in the Standard Toolbar

Whether *Help* is displayed in the [Standard Toolbar \[Seite 307\]](#) depends on the global system settings and cannot be influenced by the developer. This applies to primary windows as well as to secondary windows. *Help* is currently displayed in the standard toolbar.

Cross-References

- [Function Keys - Overview \[Seite 267\]](#)

- Help (Menu)
- [Possible Entries \(Function\) \[Seite 499\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Open (Function)

Open (Function)

[Anderes <Objekt> \[Seite 474\]](#)

This function retrieves and opens any object on the screen. *Open* is only provided for Basis tasks (for example, *Open program*). Otherwise, [Other <object> \(Function\) \[Seite 474\]](#) has to be used.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F17; if this is already used for <i>Other <object></i> : function key can be defined freely.	Menu <Object> [Seite 574]	Application toolbar: in the block of the navigation keys	O for <i>Open</i> (default)	---




Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- Other <object> (Function)

Copy (Function)

Copy is a function at task level for copying lines, areas, or objects. The user can choose this function to copy lines, areas or objects, for example, into the clipboard or to the database. The line, the area or the object are duplicated and possibly saved under a different key or name (for example, Copy document, Copy program).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icons
The function key can be defined freely.	Menu <Object> [Seite 574] (Copy entire object), menu Edit [Seite 565] (Copy object component)	Application toolbar: in the block of the self-defined function keys Standard Toolbar: local copy	C for <u>C</u> opy	 (central copy)  (Copy object)  (Standard Toolbar: local copy)

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)


List (Function)

List (Function)

[Springen \[Seite 579\]](#)

List is a function at task level the user can choose to request and display a list.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu Goto [Seite 580]	Application toolbar: in the block of the self-defined function keys	---	

Cross-References



- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- Goto (Menu)

Delete or Delete <object> (Function)

Delete is a function at task level the user can choose to delete table entries, list entries or data records (for example, Delete program). If lines are to be deleted from tables or lists, they have to be selected first. To indicate the object to be deleted, you have to specify the corresponding function name (for example, *Delete line*).

Depending on the context, the system should display a dialog box with a confirmation prompt before the action is executed (see [Prompts in Dialog Boxes \[Seite 50\]](#)).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icons
F14 (recommended)	Menu <Object> [Seite 574] (Delete entire object), menu Edit [Seite 565] (Delete object component)	Application toolbar: in the block of the standard functions	D for <i>D</i> delete	 (Delete)  (Delete Line)



Function *Delete line* is used to delete lines in a table structure.

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)

Select/Deselect (Function)

Select/Deselect (Function)

[Verschieben \[Seite 486\]](#)

[Ankreuzfeldern \[Seite 176\]](#)

Select/deselect is a function at task level the user can choose to change the select state of individual objects.

Using the Keyboard

If the user positions the cursor on a single object of the user interface (for example, a table structure) and chooses *Select/Deselect*, the entire object changes the **select state**: If it was not selected, it is selected now; if it was selected, it is deselected now. Such an object in a table is, for example, an individual field, an entire line or an entire column. A line in this case is an entire logical line, that is, both physical lines in two-line tables.

Using the Mouse

The user positions the pointer on an object and clicks the left mouse button. Then **the entire object changes the select state**: If it was not selected, it is selected now; if it was selected, it is deselected now (see Technical Notes).

The user can then initiate a function for a selected object with the function keys, menu options or pushbuttons. For example, [Copy \(Function\) \[Seite 513\]](#), [Move \(Function\) \[Seite 532\]](#) or [Delete or Delete <object> \(Function\) \[Seite 515\]](#).



"Select Again"

There can be an optional function *Select again* which selects again all lines marked with "***". ("**" in a column before the Checkboxes (Control Element) indicates exactly those objects which have already been processed or viewed in detail.) (see [Selection Methods - Overview \[Seite 95\]](#))

Technical Notes

It is currently not possible to initiate a function by clicking the left mouse button. This function can therefore only be initiated with the keyboard.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F9 (recommended); if the function Editor or Long Text (Function) [Seite 497] is also active and therefore assigned to F9 function key can be defined freely	The function must not be included in a pull-down menu.	Application toolbar: in the block of the standard functions * Standard Toolbar: local select mode	---	 (central select/deselect)  (Standard Toolbar: local select mode)

*) As long as *Select/Deselect* is assigned to a function key; as soon as this function can be initiated by a single click with the left mouse button, it is no longer displayed in the application toolbar.

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Copy \(Function\) \[Seite 513\]](#)
- [Delete or Delete <object> \(Function\) \[Seite 515\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- Move (Function)
- [Select All, Deselect All \(Functions\) \[Seite 473\]](#)
- [Select Block \(Function\) \[Seite 494\]](#)
- [Selection Methods - Overview \[Seite 95\]](#)

Menu Bar (Function)

Menu Bar (Function)

[Abbrechen \[Seite 470\]](#)

Menu bar is a function at task level the user can choose to activate the menu bar. When the function is activated, the cursor is positioned on the first menu bar option.

The user can now go through the pull-down menus using the arrow keys and choose an item by positioning the cursor on it and pressing ENTER.

If the user does not want to choose a menu option, he or she can leave the pull-down menus and the menu bar by choosing *Menu bar* or [Cancel \(Function\) \[Seite 470\]](#). The cursor is at the same position as when the user has activated the menu bar.

Alternatively, the user can operate the menus with the keyboard.

This function is automatically active in the system.

Access

The function is to be always active.

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F10 *	The function is not assigned to a pull-down menu	The function is not displayed as a pushbutton.	---	---

*) F10 must not be used for any other function

Cross-References

- [Cancel \(Function\)](#)
- [Function Keys - Overview \[Seite 267\]](#)

Hold (Function)

[Sichern ohne Prüfen \[Seite 479\]](#)

Hold is a function at task level the user can choose to save data in accounting-related tasks (for example, *Hold document*).

Hold is a synonym for [Save Without Check \(Function\) \[Seite 526\]](#). Unlike with *Post*, the held data does not have to be checked. If a consistency check is performed, then *Post* is to be used (see [Save or Post \(Function\) \[Seite 524\]](#)).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F16 (recommended)	Menu <Object> [Seite 574]	Application toolbar: in the block of the standard functions	O for <i>Hold</i>	---

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Save or Post \(Function\) \[Seite 524\]](#)
- Save Without Check (Function)
- [<Object> \(Menu\) \[Seite 574\]](#)

New Lines (Function)

New Lines (Function)

[Einfügen bzw. <Objekt> \[Seite 521\]](#)

The function *New lines* adds a new page with blank lines to a table or list. The upper line of this page is the last line of the table which contains a line.

After pressing ENTER, the blank lines are deleted from the table.



If any number of blank lines is to be inserted **within** the table, the user then chooses [Insert Many \(Function\) \[Seite 534\]](#). For inserting single blank lines before the cursor position, you can use the function [Insert or Insert <object> \(Function\) \[Seite 498\]](#).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined functions	N for <i>New lines</i> (default)	---


Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Insert Many \(Function\) \[Seite 534\]](#)
- Insert or Insert <object> (Function)
- Tables - Maintenance

Check (Function)

Check is a function at task level the user can choose to check the current object.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu <Object> [Seite 574] (function group <i>Other functions</i>)	Application toolbar: in the block of the self-defined functions	---	

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)

Reset (Function)

Reset (Function)

Reset is a function at task level the user can choose to cancel entries. *Reset* cancels entries (before pressing **ENTER**) and the fields are made ready for input again. No confirmation prompt is issued and no screen change occurs. Entries can only be cancelled on the currently displayed screen.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined function keys	---	---

Technical Notes

To avoid that incorrect data is checked, you have to provide the function *Reset* with a function which is specified as of category "E" in the Menu Painter. In addition, first define a MODULE x AT EXIT-COMMAND in the processing logic of the screen at the event PAI. This module will be run through before all checks of the DYNP so that you have control and can carry out *Reset* despite incorrect data and the further modules of PAI no longer have to be processed.

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)

Close (Function)

[Drucktastenleiste \[Seite 714\]](#)

In the SAP System, you can only close secondary windows with this function. It can be used, for example, for closing dialog boxes which display only information messages.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
ENTER key * (closing in dialog boxes)	The function is not assigned to a pull-down menu	Dialog box: in the Application Toolbar (Control Element) [Seite 227]	---	---

*) The function is very often assigned by default to the ENTER key which has the initiating focus.

Cross-References

- Application Toolbar (Control Element)
- [Modal Dialog Box \[Seite 65\]](#)
- [Function Keys - Overview \[Seite 267\]](#)

Save or Post (Function)

Save or Post (Function)

Save or *Post* (for example, of a program, order or master record) updates data in the database and performs a consistency check.



Depending on the task, *Save* or *Post* can be used for this function. If no consistency check is performed when saving or posting, then the function [Save Without Check \(Function\) \[Seite 526\]](#) or [Hold \(Function\) \[Seite 519\]](#) is to be used.

Save or *Post* can have different follow-up screens, depending on the task:

- **the same screen**

The user remains on the current screen and can continue the processing.

- **Navigation to a higher hierarchy level node**

For example, if an object component can be saved/posted individually, and another object component is processed afterwards.



In the Menu Painter, you navigate to the overview of all pull-down menus after saving the actions contained in a pull-down menu.

- **Navigation to the initial screen**


If another entire object is to be processed after the saving or posting.



Posting an invoice verification document.

In this case, the initial screen can also be processed in the background so that you directly navigate to a new entry screen after *Save* or *Post*.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F11 and CTRL + S	Menu <Object> [Seite 574]	Standard Toolbar [Seite 307] (position is determined automatically)	S for <u>S</u> ave; for <i>Post</i>	 (Standard Toolbar)

Cross-References

- [Function Keys - Overview \[Seite 267\]](#)
- [Hold \(Function\) \[Seite 540\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Save Without Check \(Function\) \[Seite 479\]](#)
- [Standard Toolbar \[Seite 307\]](#)

Save Without Check (Function)

Save Without Check (Function)

[Merken \[Seite 540\]](#)

Save without check is a function at task level the user can choose to save data without checks being performed. When the user chooses this function *Save*, the system does not have to check the held data. *Save without check* is a synonym for [Hold \(Function\) \[Seite 519\]](#). As compared to *Hold*, it is used every time the function *Save* is also used (for example, *Save screen without check*). If a consistency check is performed, [Save or Post \(Function\) \[Seite 524\]](#) is to be used.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F16 (recommended)	Menu <Object> [Seite 574]	Application toolbar: in the block of the standard functions	---	---

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Hold \(Function\)](#)
- [Save or Post \(Function\) \[Seite 524\]](#)

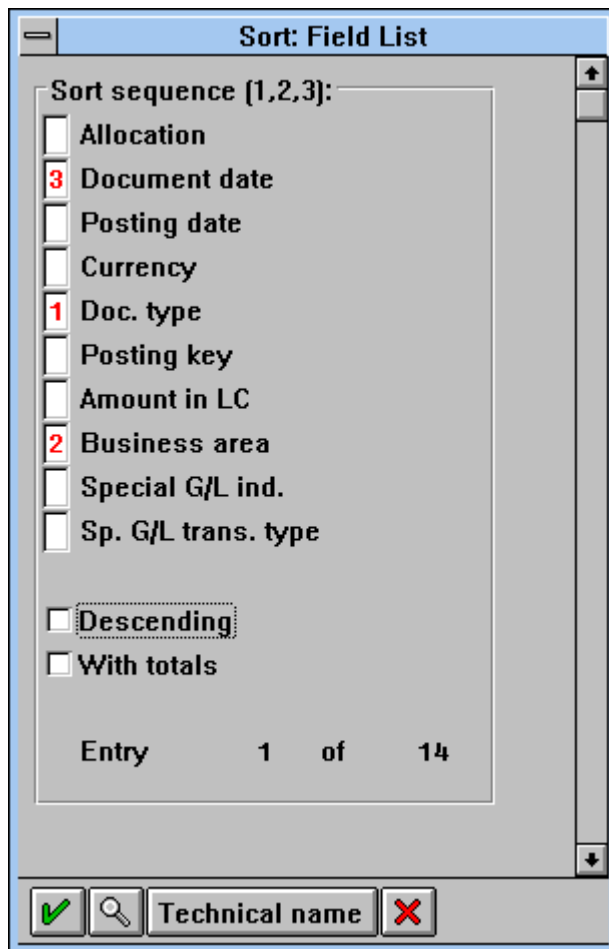
Sort (Function)

[Merken \[Seite 540\]](#)

[Sichern ohne Prüfen \[Seite 479\]](#)

This function is used for sorting the lines of a table structure. First, a dialog box is displayed in which the user can specify the criteria according to which the table structure is to be sorted. In this case, the user can determine the sequence of the sort criteria by assigning numbers for the required criteria in fields provided. The entries are sorted only after the user has confirmed the specifications by pressing ENTER and the first page of the table structure is displayed.



Illustration



Sort Sequence in a Dialog Box

Sort (Function)

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icons
F16 (recommended); if, however, Hold (Function) [Seite 519] or Save Without Check (Function) [Seite 526] is also active and therefore F16 is used: function key can be defined freely.	Menu Edit [Seite 565]	Application toolbar: in the block of the standard functions	S for <u>S</u> ort	 (ascending)  (descending)

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- Hold (Function)
- Save Without Check (Function)

Cancel/Reverse (Function)

Cancel/Reverse is a function at task level the user can choose to cancel data which has already been saved in the database.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu <Object> [Seite 574]	Application toolbar: in the block of the self-defined function keys	---	---

Cross-References


- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)

Total (Function)

Total (Function)

Total is a function at task level the user can choose to total items, amounts, and so on. The procedure used in this case is not standardized further.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely.	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined function keys	---	

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)

<Object Component> up/down/left/right (Functions)

The following functions are functions at task level the user can choose to navigate in hierarchies.

<OC> left (F17)

This function goes one object component to the left within a hierarchy (for example, "Cost center left").

<OC> up (F18)

This function goes one object component up within a hierarchy (for example, "Cost center up").

<OC> down (F19)

This function goes one object component down within a hierarchy (for example, "Cost center down").

<OC> right (F20)

This function goes one object component to the right within a hierarchy (for example, "Cost center right").

Access

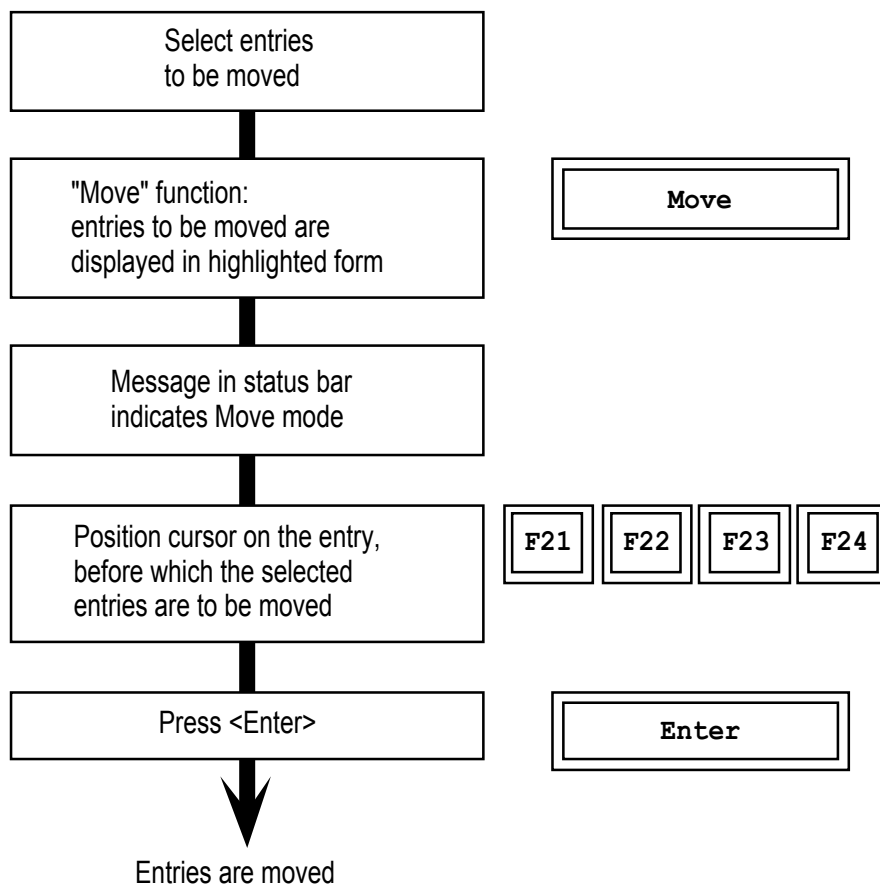
Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F17-F20	Menu Goto [Seite 580]	Application toolbar: in the block of the navigation functions	---	---

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Goto \(Menu\) \[Seite 579\]](#)


Move (Function)**Move (Function)**

Move is a function at task level the user can choose to move list or table entries. This function is used in table structures for moving lines without involving the clipboard. The following steps have to be carried out:



After the system has completed the action, the first moved line is displayed as the first line on this page.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely.	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined functions	M for <u>M</u> ove	

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)

Insert Many (Function)

Insert Many (Function)

[Einfügen bzw. <Objekt> einfügen \[Seite 521\]](#)

[Tabellen - Pflege \[Seite 383\]](#)

The function *Insert many* inserts a new page with blank lines in the table into which data can be entered. The upper line of this page is the line of the table that precedes the cursor line; the line on which the cursor was positioned is the last line of the new empty page.

If the user confirms the data input by pressing ENTER, the remaining blank lines are deleted and the table is contracted.

Instead of pressing ENTER, the *Insert many* function can be called again to insert another empty page.



[Insert or Insert <object> \(Function\) \[Seite 498\]](#) is available for inserting individual blank lines before the cursor position. [New Lines \(Function\) \[Seite 520\]](#) is used for appending an empty page to the end of the table.

In editor tasks, the name *Insert blank lines* may also be used instead of [Insert Many \(Function\) \[Seite 534\]](#).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
Function key can be defined freely	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined functions	I for <i>Insert many</i> (default)	---

Cross-References



- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- Insert or Insert <object> (Function)
- New Lines (Function)

Previous Page, Next Page (Functions)

With the functions *Previous page*, *Next page*, the user can scroll back and forth a full page in a list or a table. Each of these functions can be executed repeatedly until the beginning or end of the table structure is reached. A repeated use of the function on the first and/or last page does not result in a further scrolling.

Other functions for scrolling in table structures are [First Page, Last Page \(Function\) \[Seite 505\]](#) (F21) and [Last Page \(Function\) \[Seite 505\]](#) (F24).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icons
PAGEUP (<i>Previous page</i>), PAGEDOWN (<i>Next page</i>).	The function is not assigned to a pull-down menu.	Standard Toolbar [Seite 307] (position is determined automatically)	---	 (Previous Page)  (Next Page)

Technical Note

The function module SCROLLING_IN_TABLE is to be used for scrolling in a table structure. Among other things, the starting index and the function code for the required scrolling function are transferred to this function module. The scrolling itself is not performed by this function module. It only determines the index of the first table line to be displayed.

Cross-References

- [First Page, Last Page \(Function\) \[Seite 505\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Standard Toolbar \[Seite 307\]](#)
- [Tables - Scrolling \[Seite 376\]](#)

Previous Screen, Next Screen (Functions)

[Springen \[Seite 579\]](#)

[Erstes Bild \[Seite 523\]](#)

[Letztes Bild \[Seite 523\]](#)

[Springen \[Seite 579\]](#)

Previous screen, *Next screen* are two navigation functions at task level the user can choose to navigate within a fixed screen sequence. If the user is not guided through a fixed screen sequence, the required screens can only be accessed explicitly by specifying the navigation options in the menu [Goto \[Seite 580\]](#) or the corresponding function keys.



Other functions for scrolling in fixed screen sequences are [First Screen, Last Screen \(Function\) \[Seite 507\]](#).

In a screen sequence, [Continue \(Function\) \[Seite 540\]](#) usually navigates to the *Next screen*.

Next screen, however, is not redundant, because on one screen you might only want to carry out validations with the ENTER key or because there is a table on one screen of the sequence and the user is not to leave this table again with *Continue*. In these cases, the user can only get to the next screen by using *Next screen*.

Previous screen has no equivalent in any other function.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icons
The function key can be defined freely (but do not use the function keys F21 through F24)	Menu Goto	Application toolbar: in the block of the self-defined function keys		 (Previous screen)  (Next screen)

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Continue \(Function\) \[Seite 540\]](#)
- First Screen, Last Screen (Function)
- [Function Keys - Overview \[Seite 267\]](#)
- Goto (Menu)
- [Navigation Functions - Overview \[Seite 460\]](#)

Previous <object component>, Next <object component> (Functions)

Previous <object component>, Next <object component> (Functions)

[Markieren/Entmark. \(Funktion\) \[Seite 516\]](#)

[Springen \[Seite 579\]](#)

With the functions *Previous <object component>* or *Next <object component>*, the user can navigate back and forth between the fullscreen display of entries of a higher-level list or table. If the individual items of the higher-level list are documents, for example, then the functions are called *Previous document* and *Next document*.

If the user selects a subset of lines in the list/table (see [Select/Deselect \(Function\) \[Seite 516\]](#)), the functions only navigate between these selected lines.

Constant Retention of the Original Object Component View

If the user activates an object component with [Choose \(Function\) \[Seite 487\]](#) from the list/table and this screen contains further screens for the object component he or she can view, the original view displayed for the object component chosen at the beginning is displayed with *Previous <object component>*, *Next <object component>*.

Exception: If there are contents-related reasons, the current view can be maintained after *Previous <object component>*, *Next <object component>*.

List/Table as Further Object Components

If, after choosing an object component from the list or table, this object component again consists of a list/table the user can use to activate a "sub-component of object" screen, then *Previous <object component>*, *Next <object component>* only navigates to the current level.

Navigating to the First and Last Object Component



If the user chooses *Previous <object component>* or *Next <object component>* on the first or last displayed object component of the list/table, the developer can choose between two procedures, depending on the application:

- The higher-level list/table is again displayed to the user.
- The user remains on the screen; a message is displayed, indicating that no other previous/next object components can follow (for example, "*You have reached the last <object>*").

The user can view a particular object component with the function [Other <object component> \(Function\) \[Seite 476\]](#).

Previous <object component>, Next <object component> (Functions)

Access

Function Key	Pull-Down Menu	Pushbutton	Access Characters	Icons
F18 (<i>Previous <object component></i>), F19 (<i>Next <object component></i>) *; not active without the respective function keys.	Menu Goto [Seite 580] **	Application toolbar: in the block of the navigation keys (you use "<object> -" and "<object> +" as text for the pushbuttons)	P for <i>P</i> revious <object component>, N for <i>N</i> ext <object component>	 (Previous <object>)  (Next <object>)

*) F18/F19 may be used for other functions, if *Previous <object component>*, *Next <object component>* are not used in the task

**) If the menu *Goto* is divided up into several menus (for example, into "Header" and "Position"), the functions are contained in the pull-down menu belonging to the corresponding object components.

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- Select/Deselect (Function)
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- Goto (Menu)
- [Other <object component> \(Function\) \[Seite 476\]](#)

Continue (Function)

Continue (Function)

[Nächstes <Teilobjekt> \[Seite 515\]](#)

[Nächstes <Teilobjekt> \[Seite 515\]](#)

Continue is a provisional name for the function covered by the `ENTER` key. This function can navigate to a predefined screen or perform a check. This is a standard function assigned to the `ENTER` key.

No "Undefined" Navigation Options

Continue should only provide navigation options available through other functions. This means that other functions bring the user to the same screen as *Continue*.

Do **not** allow any "undefined" navigation options in your task.

Consistent Use

Continue can stop on a screen to perform a check.

To use *Continue* this way, repeated use of *Continue* should perform this check. The check is **not** to be performed when the user presses *Continue* the first time and leaves the screen after having pressed *Continue* a second time. Use the [Previous Screen, Next Screen \(Functions\) \[Seite 536\]](#) to navigate to another screen or a separate function for the check.

Forward Navigation

If *Continue* is not only to perform checks, it should navigate to the "next" screen. Depending on the current dialog situation, you can use *Continue* differently. However, try

- to follow the guidelines listed under Recommended Navigation Options,
- to implement the checks or the navigation to the next screen consistently on the `ENTER` key.

Recommended Navigation Options

The following navigation options for *Continue* are recommended:

- navigation from the initial screen to the screen following the initial screen.
- navigation to the next screen in a fixed screen sequence.

However, *Continue* should stop at the end of the sequence to prompt an explicit [Save or Post \(Function\) \[Seite 524\]](#).

As an alternative, after pressing *Continue*, a dialog box can automatically be displayed on the last screen in which you can specify whether you want to save/post the document or not.

- navigation through the "full screen" display of the elements of a higher-level list/table, instead of using [Next <object component> \[Seite 538\]](#).
- navigation on the last "full screen" to the higher-level list/table or stop on the last screen, depending on the alternative you have chosen for the function Next <object component>.
- *Continue* should **not** replace the function [Choose \(Function\) \[Seite 487\]](#).



Follow-Up Functions, if *Continue* Stops on the Screen

Even if the user has already pressed ENTER on a screen (function *Continue*) and thus remains on the screen due to this function (check), all functions should work "normally". It is not clear to a user why the action executed after pressing *Continue* (and thus stopping on the screen) is different from the function initiated without previously pressing *Continue*.

Continue in a Dialog Box

ENTER in a dialog box initially executes the default function defined by the developer. This default function can, for example, be called *Continue* or [Close \(Function\) \[Seite 523\]](#).

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
ENTER key	The function is not assigned to a pull-down menu.	Standard Toolbar [Seite 307] (position is determined automatically)	---	 (Standard Toolbar)  (Dialog box)

Technical Note

The function covered by the ENTER key is not explicitly to be assigned to ENTER in the Menu Painter, that is to say, you should not and may not provide a function code for it. (In case of warnings, the user would not be able to confirm the warning, because the warning is to be confirmed by pressing ENTER; the desired function code "Blank", however, is not generated but the code covered by ENTER).

Cross-References


- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Choose \(Function\) \[Seite 487\]](#)
- [Close \(Function\) \[Seite 523\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Previous Screen, Next Screen \(Functions\) \[Seite 536\]](#)
- Previous <object component>, Next <object component> (Functions)
- [Standard Toolbar \[Seite 307\]](#)

Continue (Function)

Undo (Function)

Undo is a function at task level the user can choose to cancel actions. The action to be cancelled should be the action carried out last. If it is technically possible, you can also provide the undoing of several (last) actions.

Access

Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
The function key can be defined freely	Menu Edit [Seite 565]	Application toolbar: in the block of the self-defined function keys	U for <u>Undo</u>	

Cross-References

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Function Keys - Overview \[Seite 267\]](#)

Back (Function)

Back (Function)

[Navigation zwischen Bildern \[Seite 106\]](#)

[Springen \[Seite 579\]](#)

Back is a navigation function which navigates to the previous level in the structure of a task. *Back* navigates to the previous, higher level in the hierarchy of a task structure. You must indicate to the user, for example in the title bar, that navigation takes place within a hierarchy.

Back includes checks. If an error occurs on a screen, the system displays an error message and the user cannot leave the screen.

Exception: If possible, neither a check for incorrect data is performed on the initial screen when using *Back* nor a confirmation prompt is issued, unless the task only consists of the initial screen.

No Data Loss

Back should generally not "lose" the data entered between the present screen and the navigation option.

If unsaved data may be lost, the system should display a confirmation prompt to inform the user about the critical situation. This should also allow the user to save the data or to cancel the request (see [Prompts in Dialog Boxes \[Seite 50\]](#)). If the user decides not to save the data, the data is lost.



If the user has completed the initial screen and initiated *Back*, the previously displayed data will **not** be displayed (except for SPA/GPA parameters) when displaying the initial screen later.

Possible Navigation Conflicts

If, within a task, the user can navigate to a screen from two (or more) positions that rank directly "above" the screen, then *Back* can, of course, only return to one of these locations.

This problem particularly occurs, if the navigation option is predefined as specified in the guidelines. For example, if one of the *Back* functions from a displayed object component navigates to a higher-level list/table.

It should be considered carefully, whether

- the navigation option returning one step in the hierarchy or
- the navigation option going to the previous screen

should be offered with regard to the *Back* function. The navigation option the user most likely expects should be implemented with *Back*. The remaining option should be assigned to another function.


As an alternative, you can name one function *Back to <x>* and the other *Back to <y>* if there are two (or more) navigation options that return to similar positions in the hierarchy.

Access

The function should always be active unless this is not possible for technical reasons.

Exception: In dialog boxes the function *Back* is not available.

Back (Function)

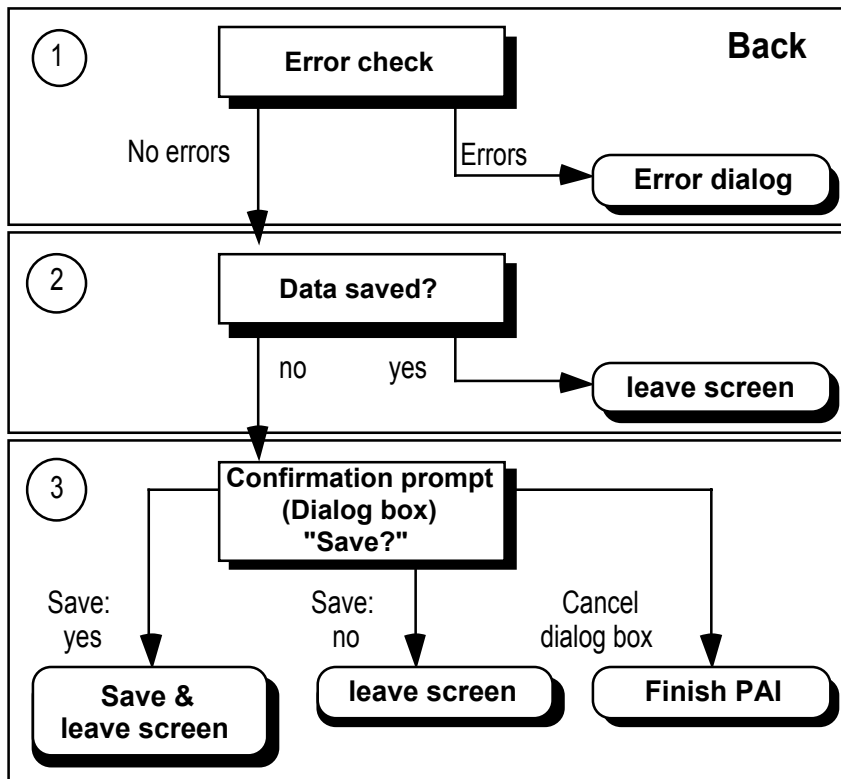
Function Key	Pull-Down Menu	Pushbutton	Access Character	Icon
F3 *	Last option in the menu Goto [Seite 580]	Standard Toolbar [Seite 307] (position is determined automatically)	B for <u>B</u> ack	 (Standard Toolbar)

*) F3 must not be assigned to any other function

Exception: If there are two or more navigation targets, one of the *Back* functions should be assigned to F3 (e.g. *Back to <x>*), the other to any other function key.

Technical Notes

If the user chooses *Back*, the entered data is processed first. If the system detects an error during the check, it initiates an error dialog.



Exception: On the initial screen, no checks for incorrect data should be performed. You have to provide the function *Back* with a function code specified as category "E". In addition, first define a MODULE x AT EXIT-COMMAND in the flow logic of the screen at the event PAI. This module will be processed before all checks of the DYNP are performed. This way, the user has control and can execute "Back" although there is incorrect data.

Back (Function)**Cross-References**

- [Application Toolbar \(Control Element\) \[Seite 227\]](#)
- [Goto \(Menu\)](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Navigation Between Screens \[Seite 106\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Standard Toolbar \[Seite 307\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)

Menus - General Guidelines and Overview

This section describes the general design of menus and the structure of the menu bar at application and task level.

[Menus - General Structure \[Seite 548\]](#)

[Menu Bar at Application Level \[Seite 555\]](#)

[Menu Bar at Task Level \[Seite 559\]](#)



[Overview of Menu Bars \[Seite 679\]](#) provides an overview of the menu bars at the individual levels of the SAP System, in particular the menu bar at task level.

Menus - General Structure

The overall functionality of the SAP System is divided into three levels (see [Levels in the SAP System \[Seite 72\]](#)). Each of these levels has a special menu structure.

This section discusses general design guidelines for menus used at all three levels.

[General Design Guidelines \[Seite 549\]](#)

[Structure and Access \[Seite 552\]](#)

[Technical Notes \[Seite 553\]](#)

[Cross-References \[Seite 554\]](#)

General Design Guidelines

[Erste Seite \[Seite 505\]](#)

[Vorige Seite \[Seite 535\]](#)

[Blätterleisten \[Seite 707\]](#)

Menus as the Basis of the Interaction

With few exceptions (see below), the user must be able to call all functions of a task (= application functions) and therefore all functions assigned to function keys from the pull-down menus.

If all application functions are displayed in menus, the user is not required to remember the commands (which is very difficult for some people) but only needs to recognize them (which is a lot easier). This way, the user can use the available system functionality more efficiently.

Functions Not in Menus

There are some exceptions to above mentioned guidelines. The following functions never appear in pull-down menus:

- F10: Navigation to the menu bar (see [Menu Bar \(Function\) \[Seite 518\]](#))
To choose a menu option, the user has to be in the menu bar.
- F9: [Select/Deselect \(Function\) \[Seite 516\]](#)
- F21–F24: [First Page, Last Page \(Function\) \[Seite 505\]](#), [Previous Page, Next Page \(Functions\) \[Seite 535\]](#).

These basic interaction functions are used very frequently and, as a result, are learned fast. They would therefore use up space in the menu without actually being needed. In addition, the functions can be more directly be initiated by the user. With graphical user interfaces, these functions are therefore controlled by interaction mechanisms which allow a more direct and quicker execution (for example scroll bars, scroll arrows in the [Standard Toolbar \[Seite 307\]](#)).

Choosing Menu Options to Execute Functions

The function selection via menus follows the object/action approach: First, the user has to specify the object class or transaction class, and then the corresponding action (see [Object-Oriented User Interface \[Seite 129\]](#)).

The object to be processed is generally in the menu bar and the actions that can be performed on it are on the first level of the pull-down menu.

If the object is in the pull-down menu, you can write the action on the same level directly behind the object but only for few actions. If the action explicitly results from the context of the object selection, you can omit the specification of an action.

Depth vs. Width of Menu Structures

The total quantity of functions for a menu bar option should be arranged on as few menu levels as possible.

General Design Guidelines

You should use a deep, tree-like structure only if this helps to make the business logic of the application more transparent. This also supports the user in choosing alternatives. However, grouping related functions generally results in a quicker interaction and a better distinction with regard to the contents of options displayed in parallel. Furthermore, menus with a flat hierarchy speed up interaction if the user only work with keyboard identification codes.

This guideline is completed and modified by the two following rules. They are to be used together.

Quickly Accessing Frequently Used Functions

If possible, the user should be able to choose the most important and most frequent functions on the first level of the pull-down menu.

In some cases, there will be many options in a menu which then to a certain extent would have to be moved to the second or third level. If there is still space for further options in the menu bar, such a menu can be split into two titles of the menu bar with less options on the first level. For the two new menu titles, you should choose different terms that are typical for the subordinate menu options.

Hiding Rarely Used Functions

Functions that are rarely used or complex functions that are only used by few people, should be "hidden" on a second or third level.

Upper/Lower Case

Menu options can consist of several words which are to be written in lowercase letter, except for the very first letter of the menu title which is always capitalized.

Status of the Menu Bar Options

Menu bar options that contain at least one option in the pull-down menu initiating a function in the current dialog state, are to be displayed as being active (generally black). If, on the other hand, the pull-down menu does not contain an option initiating a function in the current dialog state, the respective menu bar option is to be displayed as being inactive (dimmed, generally gray). The pull-down menu is opened anyway. (see Technical Notes)

When the menu title appears gray, it indicates that all functions of this menu cannot be chosen currently. However, to make the system functionality as transparent as possible and to keep the interaction with the menu bar constant, the user can nevertheless open the pull-down menu upon request. This increases the feeling of controlling the system and therefore its acceptance and reduces the number of interaction rules to be learned.

Consistency of the Menu Bar

The menu bar should be consistent within a task.

All menu bar options of the data screens should also be included on the initial screen. This avoids any inconsistency caused by the initial screen having an additional, unclear status between application level menu and task level menu.

Status of the Pull-Down Options

- An option in the pull-down menu is to be displayed as being **active** if it initiates a function in the current dialog state or if the subordinate level contains at least one active option.

General Design Guidelines

- Options in the pull-down menu which should not initiate a function in the current dialog state are to be displayed as being inactive. Options are also to be displayed as being inactive if, in turn, there are only inactive options at the subordinate level of the pull-down menu. The pull-down menu subordinate to the inactive option can be opened anyway.

If a pull-down option is displayed as being inactive, the function is neither displayed as **function key** nor as a **pushbutton**.

The inactive display signals that the user is not required to open the subordinate levels of the pull-down menu because none of its functions can currently be selected. However, to make system functionality as transparent as possible and to keep the interaction with the pull-down menu constant, the user should nevertheless be able to open the pull-down menu upon request.

Consistency of the Pull-Down Menus

Functions, that cannot be chosen in individual dialog states of a task, are to be displayed as being inactive and are not to be hidden, regardless of the user authorization.

As far as possible, the pull-down menus should have the same options within a task. This also applies to the entire application, if possible.

Dimming functions the user cannot choose because a particular authorization is required, is to enable communication between users with different authorizations. Furthermore, changing an authorization does not require to change the menu.

Structure and Access

Grouping the Pull-Down Options by Using Separators

The menu options should be combined to form useful groups. The groups are to be separated from each other by a horizontal line.

Note Concerning the Following Dialog Box

If selecting a menu option displays a dialog box, this is to be indicated by placing three periods after the menu option. Do not use blanks between the option name and the first period. (see Technical Notes)

Selection Menus with Access Characters

The user should have a chance to select a menu bar option and therefore to open a pull-down menu by entering a character on the keyboard (access character). If a pull-down menu is opened, it must also be possible to choose every option by entering a character with the keyboard. For a number of standard functions, the characters are predefined. (see [Access Characters \[Seite 184\]](#) and Technical Notes)

The used character should be typical for the option. This is usually the first letter or a letter which uniquely distinguishes options with same first letter.

Highlighting the Access Character

The access character is underlined in the menu option. (see Technical Notes)

Selecting a Menu Selection Using a Key Combination

If a menu is closed, it should be possible to choose frequently used functions by using a particular key combination. This can be a function key or another key together with the `CTRL` key. The respective combination is to be right-aligned in the menu next to the option name (see Technical Notes).

Technical Notes

Status of the Menu Options

The inactive display of menu bar options and pull-down options for which you can nevertheless open a subordinate pull-down menu is currently not possible yet.

Number of Pull-Down Options

A pull-down menu can have up to 15 options (compared to nine options so far) per level.

Note Concerning the Following Dialog Box

The three periods are to be included as part of the menu option when defining the pull-down menu in the Menu Painter and are therefore a part of the option with a maximum of 20 characters. They are not created automatically.

Selecting Menus With Access Characters

The access characters are to be defined in the Menu Painter when defining the pull-down menu. During the definition, you have to make sure that the character is unique.

Highlighting the Access Character

The highlighting occurs automatically by definition.

Selecting Menus With a Key Combination

Currently, you cannot right-align the key combination next to the menu option due to technical reasons.

Cross-References

Cross-References

- Auswahlbuchstaben
- [Blätterleisten \(Bedienelement\) \[Seite 202\]](#)
- [Access Characters \[Seite 184\]](#)
- [<Application> \(Menu\) \[Seite 564\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Environment \(Menu\) \[Seite 583\]](#)
- [Extras \(Menu\) \[Seite 585\]](#)
- [Goto \(Menu\) \[Seite 580\]](#)
- [Help \(Menu\) \[Seite 570\]](#)
- [Levels in the SAP System \[Seite 72\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Object-Oriented User Interface \[Seite 129\]](#)
- Scroll Bars
- [Standard Toolbar \[Seite 307\]](#)

Menu Bar at Application Level

The overall functionality of the SAP System is divided into three levels (see [Levels in the SAP System \[Seite 72\]](#)). Each of these levels has a special menu structure.

The application level is the second level of the three-level system hierarchy.

The R/3 System is set up such that after the user logs on, the system goes to the main menu level. The user may define to go directly to the first screen of an application by specifying the name of the corresponding menu in the user master record.

This section discusses design guidelines for the **menu bar at application level**.

[Design Guidelines \[Seite 556\]](#)

[Technical Notes \[Seite 557\]](#)

[Cross-References \[Seite 558\]](#)

Design Guidelines

Design Guidelines

Requirements

Within an application, the user should be able to choose all related object classes or transactions - so-called object classes in the following - and the corresponding actions to start a task and therefore to process a particular object of this class.

Below, you will find a step-by-step description on how to design a menu bar.

Including All Object Classes In the Menu Bar

Structuring the menu bar, you should first try to include all object classes of the application in the menu bar. The individual actions are then assigned to these object classes in the respective pull-down menus.

Creating Super Classes

If there is not enough space in the menu bar (the menu bar can have up to nine menus in addition to the *System* and *Help* menus), the object classes should be grouped together to form super classes. These super classes should be provided with mnemonic names which function as a generic term of the subordinate object classes. The super classes should be so distinct concerning the object classes contained in them and the chosen generic terms that the user can explicitly allocate the individual object classes to the individual super classes.

The object classes are assigned to the super classes in the respective pull-down menus. In the pull-down menus of the next level, the actions are then assigned to the object classes.

Creating Useful Units

If the number of object classes of a pull-down menu is more than seven (orientation value), the object classes should be grouped together within the pull-down menu to form a maximum of seven useful units separated by lines.

Repeating The Three Steps

If there is still not enough room in the menu bar, the three-step procedure described above may be repeated several times by grouping the super classes.

Flat Hierarchy

When structuring the menus in this way, the total quantity of the pull-down options subordinate to a menu bar option should be arranged on as few hierarchy levels as possible.

Technical Notes

Selecting Menus With Access Characters

The access characters are to be defined in the Menu Painter when defining the pull-down menu. In this case, you have to make sure that the character is unique.

Highlighting the Access Character

Highlighting of the access characters occurs automatically by definition.

Cross-References

Cross-References

- [Access Characters \[Seite 184\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Object-Oriented User Interface \[Seite 129\]](#)

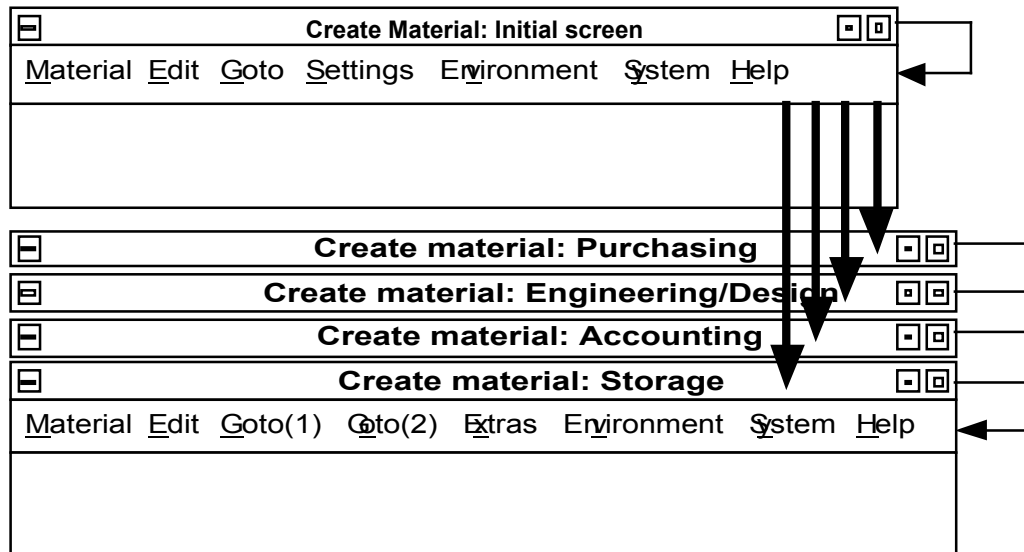
Menu Bar at Task Level

The overall functionality of the SAP System is divided into three levels (see [Levels in the SAP System \[Seite 72\]](#)). Each of these levels has a special menu structure.

The task level is the third level of the three-level system hierarchy.

This section discusses the design guidelines for the **menu bar at task level**.

Illustration



Within a task, the menu bar and pull-down menus have a consistent structure.

Within the task level of the R/3 System, the user can call a large number of individual tasks. Each of these tasks is used for processing objects of a certain object class. The menu bar is structured in the same way across all tasks so that the user can easily find his/her way around in the different tasks.

[Design Guidelines \[Seite 560\]](#)

[Cross-References \[Seite 562\]](#)

Design Guidelines

Design Guidelines

The menu bar and the pull-down menus of the SAP application functions consist of obligatory and optional entries. The **bold** typeface is used to indicate menu options that always have to be available. Entries displayed in regular typeface do not have to be available always. They are to be used and arranged in the specified form, if required. Variable entries are indicated by angle brackets.

Contents of the Menu Bar

The standard menu bar contains the following options in the specified order:

<Object> **E**dit **G**oto **E**xtras **E**nvironment <application> **S**ystem **H**elp



Phys. inventory doc. **E**dit **G**oto **E**nvironment **P**hysical inventory **S**ystem **H**elp

In addition, you can include the menu titles **View**, **Settings** and **Utilities**, if required. They are to be positioned between *Extras* and *Environment* in the specified order. Additional menus for special features of individual tasks are possible. However, you should check whether the functions possibly correspond to standard menu functions.



In addition to menus automatically added by the system, you can freely define nine entries. As a consequence, the menu bar often must extend to the next line (exception: Apple).

Defining Menu Categories

The individual menu categories are defined as follows:

<Object>

The menu <Object> [\[Seite 574\]](#) contains actions the user can choose to manipulate an object as a whole. This includes the functions for requesting, saving, printing, deleting, and closing objects.

Edit

The menu [Edit \[Seite 565\]](#) contains actions the user can choose to edit the current object component. This includes the functions for selecting, editing, cancelling, clipboard and application-specific functions.

Goto

The menu [Goto \[Seite 580\]](#) contains all object components or all screens of the current object the user can navigate to by choosing the option.

Extras

The menu [Extras \[Seite 585\]](#) contains functions the user can choose to complete the current object/object component and which are not constantly needed. They generally initiate the display of a dialog box which contains additional information/options.

Environment

The menu [Environment \[Seite 583\]](#) contains functions the user can choose to call transactions or reports of other applications.

<Application>

The menu [<Application> \[Seite 564\]](#) contains the menu options of the higher-level application area menu. It is used to accelerate cross-transaction work processes and to integrate the standard menu if accessed with path names/transaction codes (see [Accessing a Task \[Seite 77\]](#)).

View

The *View* menu contains the functions the user can choose to display the current object or object component in different views (for example, when switching between a one-line and two-line display of a table). The functions do not change the data of the object or the object component.

Settings

The *Settings* menu contains functions which allow the user to set user-specific transaction parameters.

Utilities

The *Utilities* menu contains functions which allow not only the processing of the current object but also cross-object processing. This includes general functions, such as *Delete*, *Copy*, and *Print*.

Technical Note

The menu bar is defined in the Menu Painter.

Cross-References

Cross-References

- [Accessing Tasks \[Seite 77\]](#)
- [<Application> \(Menu\) \[Seite 564\]](#)
- [Edit \(Menu\) \[Seite 565\]](#)
- [Environment \(Menu\) \[Seite 583\]](#)
- [Extras \(Menu\) \[Seite 585\]](#)
- [Goto \(Menu\) \[Seite 580\]](#)
- [Levels in the SAP System \[Seite 72\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Object-Oriented User Interface \[Seite 129\]](#)

Menus

This section lists the individual menus of the menu bar at task level in alphabetical order.

[<Application> \(Menu\) \[Seite 564\]](#)

[<Object> \(Menu\) \[Seite 574\]](#)

[Edit \(Menu\) \[Seite 565\]](#)

[Environment \(Menu\) \[Seite 583\]](#)

[Extras \(Menu\) \[Seite 585\]](#)

[Goto \(Menu\) \[Seite 580\]](#)

[Help \(Menu\) \[Seite 570\]](#)



[Overview of Menu Bars \[Seite 679\]](#) provides an overview of the menu bars at the individual levels of the SAP System, in particular the menu bar at task level.

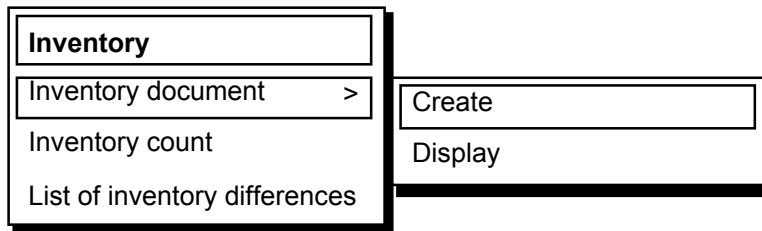
<Application> (Menu)

<Application> (Menu)

The <Application> menu is a pull-down menu at task level. It contains the menu options of the higher-level menu at application level and is to accelerate cross-transaction work processes.

This pull-down menu is **optional** and is the last menu before the *System* menu.

Illustration



Structure of the Menu <Application> (Example)

Use

The <Application> menu should be included, if possible. It allows the user to alternate between the application functions of the current application. The current application function is exited or cancelled. A confirmation prompt must therefore be issued to avoid a possible data loss (see [Prompts in Dialog Boxes \[Seite 50\]](#)).

The menu replaces the explicit exiting of the current application function and the selection of a new application function at application level. The current object class is to be displayed as being inactive because the same functionality is already available in the left menu <Object> [\[Seite 574\]](#).

All object classes of the application and its functions should be included, if possible. If this is not useful because of authorization or job description reasons and/or impossible for space reasons, the most important object classes should be listed as complete as possible.

The <Application> menu should be clearly structured. Otherwise, the user does not have a clear idea of the menu's contents and will no longer use the menu if the search has proved to be in vain.

Cross-References

- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)

Edit (Menu)

[Springen \[Seite 579\]](#)

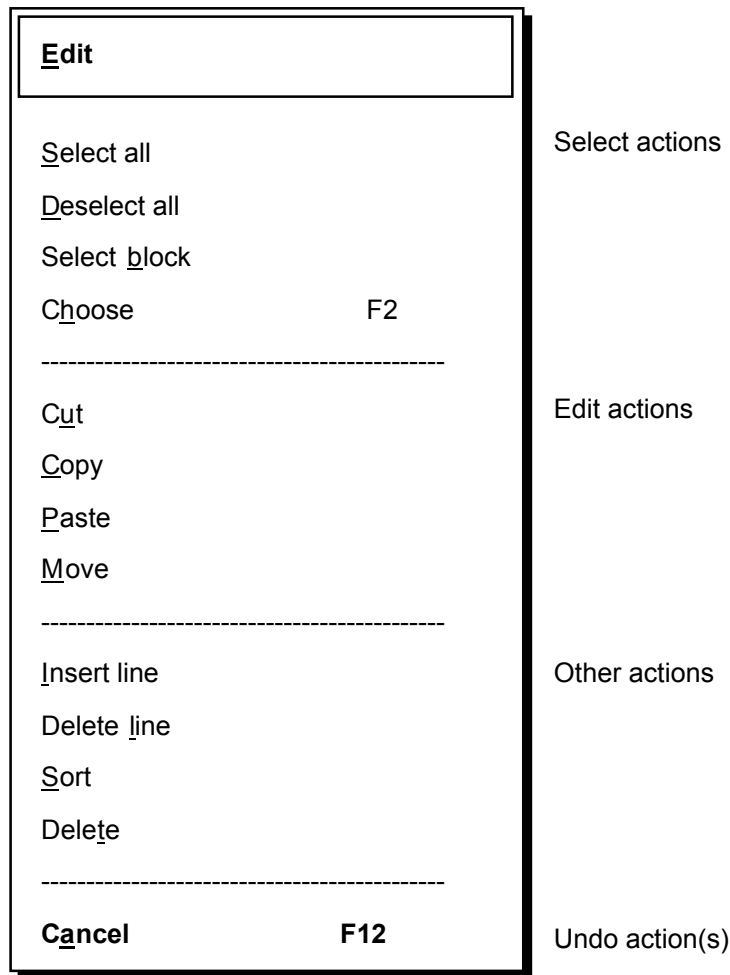
[Abbrechen \[Seite 470\]](#)

The *Edit* menu is a pull-down menu at task level. It contains the actions the user can choose to edit the current object component. This includes functions such as *Select* or *Edit*, application-specific functions and the *Cancel* function.

This pull-down menu is **obligatory** and always appears between the [<Object> \(Menu\) \[Seite 574\]](#) and the Goto (Menu).

Illustration

The only obligatory menu option is the Cancel function. It always appears as the last option in the menu.



Structure of the *Edit* menu

Edit (Menu)

[Pull-Down Options \[Seite 567\]](#)

[Cross-References \[Seite 569\]](#)

Pull-Down Options

[Alle Markieren/Alle Entmark. \[Seite 473\]](#)

[Widerrufen \[Seite 543\]](#)

[Abbrechen \[Seite 470\]](#)

The menu generally contains the following options:

Select Operations

The function *Select all* executes the selected function on all entries of an associated list. The function *Deselect all* takes back all selections in the associated list.

If the selection actions appears in a cascading menu at the second level, the term "selections" is used on the first level:

Selections ->	<u>s</u> elect all
	<u>d</u> eselect all
	Select <u>b</u> lock



The access character "D" as in *Delete* is defined for Select All, Deselect All (Functions). The deletion of selected objects, however, is something completely different as compared to the deletion of selections of the objects and you should well be aware of this fact when making your choice.

As function key names for these actions, the same names are used as for the corresponding pull-down options. If possible, they should be written in full length in the function key. If there is not enough space, they can be abbreviated as follows: *Sel.all*, *Desel.all* und *Sel.block*.

By choosing the function [Choose \[Seite 487\]](#), options are chosen from table structures, generally followed by executing a function on these options.

Edit Actions

The edit actions are to be used in accordance with the guidelines specified in [Functions - Overview \[Seite 451\]](#).

Other Actions

Other actions here refers to specific functions which affect an object component of the entire object. This covers, for example, the functions [Delete or Delete <object> \(Function\) \[Seite 515\]](#) and [Print \(Function\) \[Seite 496\]](#). If these functions refer to the object as a whole, they are to be included in the menu [<Object> \(Menu\) \[Seite 574\]](#).

Reset Actions

Functions for undoing actions (e.g. Cancel/Reverse or [Reset \[Seite 522\]](#)) are also to be placed in the *Edit* menu.

Pull-Down Options**Cancel**

Cancel (Function) always has to be active. This function cancels the processing of the current screen (object component) and returns to a defined screen. A confirmation prompt is issued if data may be lost.

Technical Note

The status of the options (that is, the active/inactive display) should be adapted dynamically to the current object component. The options within a task should generally be set to an inactive status rather than to delete them altogether.

Cross-References

- Abbrechen (Funktion)
- [Cancel \(Function\) \[Seite 529\]](#)
- [Choose \(Function\) \[Seite 487\]](#)
- [Delete \(Function\) \[Seite 515\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [<Object> \(Menu\) \[Seite 574\]](#)
- [Print \(Function\) \[Seite 496\]](#)
- [Reset \(Function\) \[Seite 522\]](#)
- [Undo \(Function\) \[Seite 543\]](#)

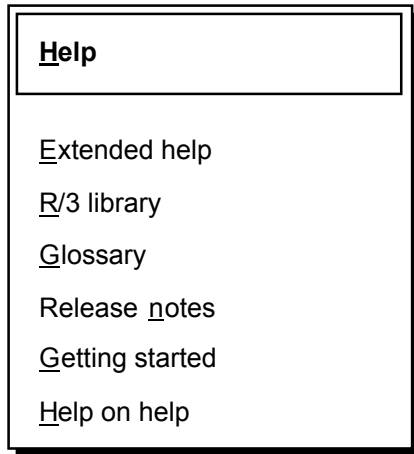
Help (Menu)

Help (Menu)

The *Help* menu is a pull-down menu which is always available in the system. It provides the user with information on his or her current situation in the R/3 System or the tasks.

This menu is always the rightmost item in the menu bar.

Illustration



Structure of the *Help* menu

[Pull-Down Options \[Seite 571\]](#)

[Technical Notes \[Seite 572\]](#)

[Cross-References \[Seite 573\]](#)

Pull-Down Options

This section explains the individual options of the *Help* menu.

Extended Help

This function displays online manuals or report documentation. If the user chooses this function on the initial screen of a program, the system displays general documentation on the current program. If the user is on the initial screen of a transaction, the system displays the contents of the corresponding online manual, if available.

R/3 Library

This function accesses the CD-ROM library with SAP documentation. This function is only available under Windows.

Glossary

This function accesses the glossary. The user receives a list of SAP glossary terms from the current application. The user can display the glossary definitions of these terms.

Release Notes

This function accesses the release notes. The user chooses the function to navigate to the SAP Information System. The user can display a complete list or choose by attributes or keywords.

Getting Started With the R/3 System

Under Windows, the user can access *Getting started with the R/3 System* online.

Help on Help

A help text is displayed which explains the different help functions of the system.

Technical Notes

Technical Notes

For menu options which consider the current context, the application developer has to take appropriate actions.

Extended Help

In Transactions

Maintenance of the documentation for *Extended help* is performed as follows:

- for the structure maintenance in transactions
 - SO70 or
 - with *Tools* → *Hypertext* → *Display*
- for text module maintenance in transaction
 - SO61 or
 - with *Tools* → *Hypertext* → *Maintain module*

A structure is assigned to the current dialog using table TTCDS. You have to specify: program name, transaction code, object name (structure), object type (documentation class) and object name. When calling *Extended help*, the help system determines the table entry for the current transaction and displays the structure specified there.

In Reports

Report documentation is created with transaction SE61 or transaction SE38.

Cross-References

- [Help \(Function\) \[Seite 509\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Possible Entries \(Function\) \[Seite 499\]](#)

<Object> (Menu)**<Object> (Menu)**

The <Object> menu is a pull-down menu at task level. It contains the actions which affect an object as a whole. This includes the functions for requesting, saving, printing, deleting, and closing objects.

This pull-down menu is **obligatory**. It is always the leftmost item in the menu bar. The name of the object that was chosen on the next higher level (application level) is used as a menu bar option.

[Illustration \[Seite 575\]](#)

[Pull-Down Options \[Seite 577\]](#)

[Cross-References \[Seite 579\]](#)

Illustration

The menus consist of obligatory and optional options. Menu options that generally always have to be available are displayed in the following in **bold** typeface. Options in regular typeface do not always have to be available. If used, they are to be used and arranged in the specified form.

On Initial Screens

<Object>		
<u>O</u>ther <object>	F17	Actions to select another object
<u>C</u>reate		
<u>C</u>hange		
<u>D</u>isplay		
Copy from...		

<u>S</u>ave	F11	Actions to save and to change modes
<u>H</u>old		
<u>G</u>enerate		
<u>D</u>isplay <-> Change		

<u>P</u>rint	F13	Other actions (e.g. import/export functions)

<u>D</u>elete	F14	Exit actions
<u>E</u>xit	F15	

The <Object> menu on initial screens

On Data Screens

On data screens, the menu can be used in the above form or in a reduced version:

Illustration

<Object>		
<u>O</u> ther <object>	F17	Actions to select another object
Copy from...		

<u>S</u> ave	F11	Actions to save and to change modes
Display <-> Change		

<u>P</u> rint	F13	Other actions (e.g. import/export functions)

<u>E</u> xit	F15	Exit actions

The <Object> menu on data screens

Pull-Down Options

[Anderes <Objekt> \[Seite 474\]](#)

[Ändern \[Seite 544\]](#)

[Anderes <Objekt> \[Seite 474\]](#)

[Sichern bzw. Buchen \[Seite 524\]](#)

[Merken \[Seite 540\]](#)

[Sichern bzw. Buchen \[Seite 524\]](#)

[Generieren \[Seite 508\]](#)

[Ändern \[Seite 544\]](#)

[Ausführen \[Seite 485\]](#)

[Bearbeiten \[Seite 683\]](#)

[Bearbeiten \[Seite 683\]](#)

[Beenden bzw. Ende <Arbeitsgebiet> \[Seite 492\]](#)

The menu generally contains the following options:

Other <Object>

The function Other <object> allows the user to process another object with the same action and navigates to the initial screen of the transaction. If the processing step of an object is interrupted by choosing a function without saving all the changes made, the system displays a dialog box. The user is prompted to save the changes in the edited object (see [Prompts in Dialog Boxes \[Seite 50\]](#)).

Create, Change, Display

The functions [Create \[Seite 479\]](#), Change and [Display \[Seite 481\]](#) allow the user to process the same or another object with another action and navigate to the initial screen of the transaction. As for Other <object>, a confirmation prompt must be issued to avoid a possible loss of data (see [Prompts in Dialog Boxes \[Seite 50\]](#)).

Copy From...

The function *Copy from...* is optional. It displays a dialog box in which information on a copy object can be entered (see [Referencing Objects \[Seite 134\]](#)).

Save

The function Save or Post should be available as often as possible. It updates the data in the database. After executing the function, the system navigates to the initial screen. The parameter settings of the last initial screen are saved.

Exception: In tasks that are similar to an editor, *Save or Post* only updates the data.

Hold

The function Hold is optional. It saves the data of an object without checking the data. *Hold* is used for all cases in which the function Post is used.

Pull-Down Options

Generate

This function [Generate \[Seite 496\]](#) is also optional. It saves the object, carries out a consistency check and generates the object (for example, *Generate screen*).

Display <-> Change

The function [Display <-> Change \[Seite 483\]](#) is optional. If used, it is only active on data screens. It initiates a change in the change or display transaction but remains on the current screen, unlike the function Change or [Display \[Seite 481\]](#). If only one direction is needed, this function is called *Display -> Change* or *Change -> Display*.

Print

The function Print is optional. It prints, for example, a list, a screen duplicate or a purchase order (for example *Print purchase order*, *Print delivery note*). If the object to be printed is not the entire object but an object component, this function is to appear in the Edit menu.

Delete

The function [Delete \[Seite 515\]](#) is optional. It appears in the menu if the user is authorized to delete the entire object. If the user is only authorized to delete object components, this function is to appear in the Edit menu.

Copy

The function [Copy \[Seite 513\]](#) is optional. It appears in the menu if the complete object is copied to the database. This happens typically in tool tasks. For example, the user copies a program or a screen.

Exit

The function Exit exits the task. It must be available and active in every task **at any time** (if even required-entry fields are filled incorrectly). If the processing step of an object is interrupted by exiting a function without saving all the changes made, a dialog box is displayed. The user is prompted to save the changes in the edited object (see [Prompts in Dialog Boxes \[Seite 50\]](#)). If the user saves the changes, the system navigates to the higher-level menu level of the application after closing the dialog box.

Technical Note

Exit

Exiting the application function for required-entry fields that are not filled is currently only possible by means of the function code "E".

Cross-References

- [Change \(Function\) \[Seite 477\]](#)
- [Create \(Function\) \[Seite 479\]](#)
- [Copy \(Function\) \[Seite 513\]](#)
- [Display \(Function\) \[Seite 481\]](#)
- [Display <-> Change \(Function\) \[Seite 483\]](#)
- [Execute \(Function\) \[Seite 485\]](#)
- [Delete \(Function\) \[Seite 515\]](#)
- [Functions - Overview \[Seite 451\]](#)
- [Function Keys - Overview \[Seite 267\]](#)
- [Hold \(Function\) \[Seite 519\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Other <object> \(Function\) \[Seite 474\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Save or Post \(Function\) \[Seite 524\]](#)

Goto (Menu)

Goto (Menu)

[Bearbeiten \[Seite 683\]](#)

[Zurück \[Seite 481\]](#)

[Nächstes <Teilobjekt> \[Seite 515\]](#)

[Voriges <Teilobjekt> \[Seite 515\]](#)

[Anderes <Teilobjekt> \[Seite 476\]](#)

[Voriges Bild/Nächstes Bild \[Seite 536\]](#)

[Zurück \[Seite 481\]](#)

The *Goto* menu is a pull-down menu at task level. It contains all object components of the current object the user can navigate to by choosing the option.

This pull-down menu is **obligatory**. It always appears directly to the right of the Edit menu as the third item in the menu bar.

Illustration

The only obligatory menu option of this menu is the function [Back \[Seite 544\]](#) which is the last option in this menu.

<u>G</u> oto	
<u>O</u> verview	
<u>H</u> eder	

<u>N</u> ext <object component>	F19
<u>P</u> revious <object component>	F18
O <u>t</u> her <object component>...	F17

<u>B</u> ack	F3

Example for the *Goto* menu

Use

The processing of an object consists of a sequence of object component processing. This menu contains all object components of the chosen object and is used for navigating to the screens on which the data of these object components can be displayed or edited. It is the navigation menu within the individual object components of an object.

The mentioned object components which are included in this menu for a direct navigation by choosing a menu option should be available for all objects of the class, if possible.

Pull-Down Options

The *Goto* menu generally contains the following options:

Overview, Header,...

Choosing one of these menu options, the system navigates to the named data screen. General SAP screens, such as the Overview screen, Detail screen or Header, are entered in the *Goto* menu with the name for the <screen> as defined in [Title Bar \[Seite 152\]](#). For all screens not included in the list of general SAP screens, a significant name has to be found which is included in the title and in the menu.

Next, Previous, Other <Object Component>...

The current object component is changed by choosing one of these options: [Next <object component> \[Seite 538\]](#) goes to the next one, [Previous <object component> \[Seite 476\]](#) navigates to the object specified in a dialog box. The functions [Previous screen \[Seite 536\]](#) and [Next screen \[Seite 451\]](#) are to be used for the relative screen change in fixed screen sequences (see [Functions - Overview \[Seite 451\]](#)).

Back

The function *Back* must always be available. When navigating within an object, it goes to a defined object component within the object from where the user can continue to work well-ordered. The current task is not interrupted. Checks are performed. A confirmation prompt must be issued to avoid a possible loss of data.

Back is the last option of the pull-down menu at first level. If the *Goto* menu is divided up into several menus (for example, in Header and Item), then *Back* is to be included in the first menu from the left as the bottom option of the pull-down menu at first level.

Cross-References

- [Back \(Function\)](#)
- [Functions - Overview \[Seite 451\]](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)
- [Navigation Functions - Overview \[Seite 460\]](#)
- [Other <object component> \(Function\)](#)
- [Previous Screen, Next Screen \(Functions\)](#)
- [Previous <object component>, Next <object component> \(Functions\)](#)
- [Title Bar \[Seite 152\]](#)

Goto (Menu)

Environment (Menu)

[Zusätze \[Seite 567\]](#)

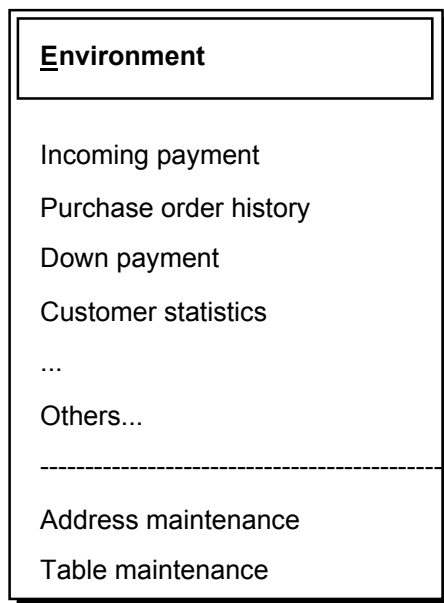
[Springen \[Seite 579\]](#)

[Zurück \[Seite 481\]](#)

The *Environment* menu is a pull-down menu at task level. It contains functions the user can choose to call tasks or reports of other applications.

This pull-down menu is **optional**. It appears to the right of the Extras menu. If the *Extras* menu is not used, the *Environment* menu appears to the right of the Goto menu.

Illustration



The *Environment* menu

Use

On the one hand, the functions in this menu should provide the user with information occasionally needed for the processing of the current object. They either call a display transaction of another application or start a report without exiting or cancelling the current processing.

On the other hand, this also includes maintenance transactions of other applications which the user occasionally needs for the processing of the objects. This integration is designed to allow an integral work process. In this case, the current object processing is also only interrupted.

Display - Maintain

The group of display functions should be separated visually from the group of maintenance transactions.

Environment (Menu)

Returning to the Task

From the called task, the user should always be able to return to the interrupted task by using a constant function key (or [Back \[Seite 544\]](#)).

Technical Note

To enable the user to return to the starting point after choosing an option from the *Environment* menu, use the statement CALL TRANSACTION (another session is opened). Do **not** use LEAVE TO TRANSACTION as it does not return to the starting point (the old session will be closed).

Cross-References

- [Back \(Function\)](#)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)

Extras (Menu)

[Springen \[Seite 579\]](#)

[Umfeld \[Seite 565\]](#)

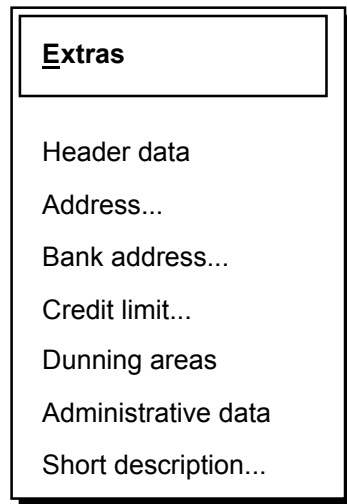
[Springen \[Seite 579\]](#)

[Springen \[Seite 579\]](#)

The *Extras* menu is a pull-down menu at task level. It contains functions the user can choose to complete the current object/object component and which are not constantly needed. These functions generally initiate the display of a dialog box which contains additional information/options.

This pull-down menu is **optional**. It appears between the [Goto \[Seite 580\]](#) menu and the [Environment \[Seite 583\]](#) menu.

Illustration



The *Extras* menu

Use

This menu contains all functions which supply additional data or additional options for the current object component or object.

The *Goto* menu contains the main screens of the transaction which generally have to be passed during the object processing but which are relatively independent from each other. The *Extras* menu, on the other hand, often contains the secondary screens of a main screen which are only occasionally processed or displayed. The information of the secondary screen depends directly on the main screen.

To make this dependency structure clear on the user interface, the calls of the secondary screens should not be arranged under *Goto* and the additional information should be displayed in a dialog box.

Extras (Menu)**Cross-References**

- Environment (Menu)
- Goto (Menu)
- [Menu Bar at Application Level \[Seite 555\]](#)
- [Menu Bar at Task Level \[Seite 559\]](#)
- [Menus \(Control Elements\) \[Seite 295\]](#)
- [Menus - General Structure \[Seite 548\]](#)

Texts

This section describes text layouts. It deals with aspects such as abbreviations, separations, the formulation of messages, use of country-specific characters, and also highlighting.

A further section discusses points to be borne in mind when translating the R/3 System into foreign languages.

[Using Texts on Screens \[Seite 588\]](#)

[Abbreviations \[Seite 593\]](#)

[Formulations \[Seite 601\]](#)

[General Messages \[Seite 604\]](#)

[Highlighting \[Seite 603\]](#)

[Hyphenation \[Seite 620\]](#)

[Message Types and Error Groups \[Seite 611\]](#)

[Translation \[Seite 621\]](#)

[Upper/Lowercase \[Seite 619\]](#)

Using Texts on Screens

Using Texts on Screens

Window Type

The following rules apply for primary as well as for secondary windows (dialog boxes). They apply for normal screens only, not for assistant screens.

Text Types

Use texts on a screen for the following text types only:

- Diagnosis texts
Example: The validation period of the entered condition overlays conditions with smaller validation periods. The system deletes these conditions when saving.
- Explanatory texts
Example: The calendar is client-independent. Every modification effects all clients.

Do not use texts on a screen for the following text types:

- Polite texts
Example: Please enter a plan variant.
Polite texts contain redundant information. They increase processing time and should not appear in dialog boxes either.
- Help texts
Example: To be able to book a document, you must first fill in the header data, then enter the invoice data and set the control flag. Finally enter the individual document positions.
- Help texts such as step-by-step instructions belong to F1 Help or Extended Help, but never on a screen.

Length, Formulation

Formulate texts on a screen in a short and unambiguous way. For a new category of information, start a new paragraph. Separate paragraphs by a blank line.

Position and Reference of a Text

You can display texts at the following positions:

- in the header area of the screen and without group box if the text contains general information on the screen,
- in the footer area of the screen and without group box if the text contains supplementary information on the screen;
- within the group box if the text contains information relevant for the group
 - at the top margin of the group box if the text contains general information concerning the group
 - preferably at the bottom margin of the group box if the text contains supplementary information on the group

- within the group box next to the elements if the text contains information relevant for the group or part of it.

If the screen contains header data, the texts should appear in the screen footer area.

Justification of the Text

1. Texts without symbol/icon

Enter texts left-justified. In the header area as well as in the footer area or the group box, start from the leftmost position.

2. Texts with symbol/icon

If you want the text to be accompanied by a symbol or icon, position it to the left of the text block. Do not place it in the middle of a text. This rule also applies for texts within group boxes or in message dialogs.

Font, Background

Present the text in the same way as field descriptions or check texts: in proportional font on the screen background.

Icons

You can use the following icons to accompany texts:

- message icons
- status icons
- the "info" icon
- icons for field descriptions (for example, date, time, phone, and so on)

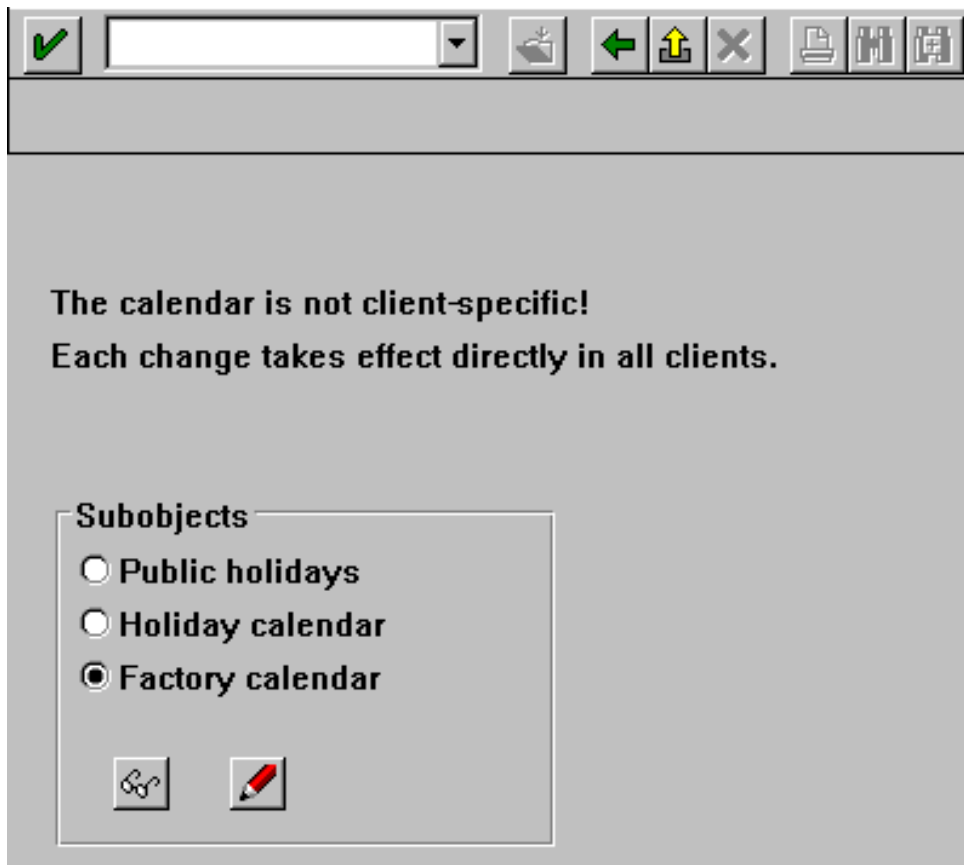
Translation

When using texts in screens, keep in mind that these texts will also be translated.

Sample Screens

Explanatory text

Using Texts on Screens



The explanatory text is positioned at the top of the screen without group box, because it contains general information that concern the entire screen.

Diagnosis text

Errors as a Result of Overlapping Validity Periods

The validity period of the condition created overlaps with conditions with shorter validity periods.

Created condition

Sales org.	Distr. chl	Material	Description
0001	01	BICYCLE	Bicycle for serial num

Valid from

17.07.1997

Valid to

31.12.9999

Overlapping conditions

Valid from	Valid to
17.07.1997	31.12.1998
17.07.2004	31.12.2009

Confirm


Delete condition

The diagnosis text is positioned at the top of the screen without group box, because it contains general information that concern the entire screen.

Diagnosis text with icon

Using Texts on Screens

Errors as a Result of Overlapping Validity Periods

 The validity period of the condition created overlaps with conditions with shorter validity periods. These conditions will be deleted when you save.




Created condition

Sales org.	Distr. chl	Material	Description
0001	01	BICYCLE	Bicycle for serial num

Valid from: 17.07.1997
Valid to: 31.12.9999

Overlapping conditions

Valid from	Valid to
17.07.1997	31.12.1998
17.07.2004	31.12.2009

 Confirm  Delete condition 

The diagnosis text is positioned at the top of the screen without group box, because it contains general information that concern the entire screen. Since the piece of information is important, the diagnosis text is accompanied by a message icon.

Abbreviations

Space restrictions may necessitate the use of abbreviations in dialog boxes, for field names in the applications, for lines in the menu bar or pull-down menus and for function key names. Use Webster's Ninth New Collegiate Dictionary as the reference for abbreviating words.

[Guidelines \[Seite 594\]](#)

[Examples for the ABAP Dictionary \[Seite 597\]](#)

[Technical Notes \[Seite 599\]](#)

[Cross-References \[Seite 600\]](#)

Guidelines

Guidelines

Using Abbreviations

Do **not** use abbreviations if sufficient space is available or if the user rarely accesses a particular screen.

Using Accepted Abbreviations

If there is already an accepted abbreviation for a word, use it. Use abbreviations only if you are sure that the user can easily understand them. In general, do not use abbreviations that consist of only one letter.



Good: Dr.; Su.; min.; DM
Bad: Dctr; Snday; minmum; Deu. Mark

Explanation: Most abbreviations are known to the user from his/her own work area or through everyday language and not from DP or SAP jargon. Create abbreviations in cooperation with the users.

Creating New Abbreviations

If there is no accepted abbreviation available, words can be abbreviated in accordance with the following guidelines arranged in order of priority.

1. Omitting the end of the word:

If possible, abbreviations for individual words should be created by omitting the end of the word. Menu options and function key names **should** have a period at the end of the word.

Reason: If choosing a menu option or function key results in a dialog box, three periods are usually added to the function name. The abbreviation of a function name can therefore only be recognized here if four periods follow the function name.

In a table column heading, the period should also be used, if possible.



"abbreviation" becomes "abb."
"date" becomes "dat."
"costs" becomes "cost."

2. Omitting vowels:

Vowels should only be omitted if you cannot form a meaningful word fragment. Vowels that do not influence the pronunciation of the word very much can be left out.



"profession" becomes "prfssn."
"group" becomes "grp."

3. Creating an acronym:

Acronyms (artificially created new terms consisting of the first letters of the word fragments or individual words) should only be created if this is clear from the application field (for example, GR for goods receipt in Purchasing) or if this is required for table column headings.

If acronyms are used, they should be included in a list for an application area. This list must be regularly updated and distributed to the other applications. You should always use existing acronyms for other abbreviations of words which contain the same word fragment.



"Data Processing" becomes "DP"

"Human Resources" becomes "HR"

"General Ledger" becomes "GL"



Never use a closing period if the last letter of a word is contained in the abbreviation.

For example, "screen" becomes "scrn". The word fragment "number" is generally no longer used in the field name.

Abbreviating Two-Word Structure

The guidelines for creating a new abbreviation apply here as well.

The individual word fragments are **not** separated by a period. Each new word fragment always begins with a capital letter if the previous word fragment has been abbreviated.



"Start date" becomes "StrtDate"

"Document type" becomes "DocType"

"Tolerance key" becomes "TolKey"

Abbreviating Several Words

Proceed as follows when abbreviating several words:

1. If the previous word was abbreviated according to the guideline "Omitting The End of the Word", it should end with a period and be followed by a blank, if there is enough space.



"Fixed vendor" becomes "Fix. vendor"

or even "Fix.vendor"

"Next object" becomes "Nex. object"

or even "Nex.object"

"Fixed capacity requirement" becomes "Fix. CapReq."

Guidelines

or even "Fix.CapRep."

2. If the last letter of the word is used, insert a blank between the words.



"Create new charge" becomes "Crte new chrg."

"Fixed vendor" becomes "Fixed vend."

3. If the second or a following word is written in lowercase, use the lowercase.



"Fixed date" becomes "Fix. date"

Choosing From Several Words

If you look at the short descriptions in the ABAP Dictionary, you can also find complete phrases, such as

"Fixed machine-related capacity requirements in hours".

For the abbreviation, you should only choose the significant words. You may need to change the word order to form a suitable abbreviation. The guidelines 1. to 3. apply for the creation of abbreviations.



"Fixed machine-related capacity requirements in hours" becomes "Mach.fix" or "Mach. fix"

Compounds

In the case of different abbreviations for words which consist of more than two word fragments, you should make sure that individual word fragments are not completely omitted. Otherwise, it could result in the impression that two different things are meant. For example, "Minimum order amount" could be abbreviated to "MinAmount" and also to "MinOrdAmount". Here, it is better to use either "MinOAmt" or "MinOrdAmount", so that the word fragment "order" is not forgotten.

Examples for the ABAP Dictionary

The following examples are listed in ABAP Dictionary sequence. Here, you can define field names in four different lengths: short - medium - long - header (see Technical notes). The length of the output field is given in parenthesis.

Transaction Type (Inventory Management)		
Trans.type (10)	Transaction type (20)	
Transaction type (15)	TType (5)	
Object Group Number		
ObjGrp. (10)	Object group (20)	
Object group (15)	ObjGrp. (7)	
Fixed Vendor		
Fixed vendor (10)	Fixed vendor (20)	
Fixed vendor (15)	F.ve. (6)	
Fixed Operator-Related Capacity Requirements in Hours		
Fixed pers (10)	Fixed personnel (20)	
Fixed personnel (15)	F.pers (5)	
Invoice Receipt Quantity		
IR quantity (10)	IR quantity (20)	
IR quantity (15)	IR qu. (6)	
Tolerance Key		
Tol. key (10)	Tolerance key (20)	
Tolerance key (15)	Tol. (4)	
Formula Number Unit Costing		
Formula no. (10))	Formula number (20)	
Formula number (15)	FNo (3)	
Volume Variance on Order		
Vol.var. (10)	Vol.var. Order (20)	
Vol.var.order (15)	VVarO. (7)	
or if there is not enough space: VVarO (6), that is, the period may also be omitted in headers.		
Number of Price Components		
No.PrC. (10)	No. price comp. (20)	
No.price comp. (15)	Pc (2)	

Technical Notes

ABAP Dictionary Keywords

In the ABAP Dictionary, four field names of different length have already been predefined:

- Short keywords, length 10 (field name)
- Medium keywords, length 15 (field name)
- Long keywords, length 20 (field name)
- Header, maximum length 55 (table headings)

Field names for table headings are entered under header. The length of the header is determined by the length of the output field.

Field names before input/output fields are entered under keywords. The basic guideline is that the entered words should always be two letters shorter than the displayed or entered length. In this case, you should whenever possible use the proposed lengths to facilitate the translation.

The difference between the length entered in the ABAP Dictionary and the length of the field name itself may be smaller than two in cases where the word can be written out and where you cannot abbreviate usefully by omitting two letters (this particularly occurs in case of the 10-character keywords).

Cross-References

Cross-References

- [Field Names \[Seite 324\]](#)
- [Formulations \[Seite 601\]](#)
- [Upper/Lowercase \[Seite 619\]](#)

Formulations

To provide the user with a work environment that is easy to understand, guidelines have also been established for the formulation of texts. These guidelines are to be observed on the different levels, for example, when choosing terms, when formulating pull-down options consisting of object and an action or when formulating the different messages.

Guidelines

Terms

If possible, use terms which the user knows and which he/she can understand.

Suffixes

Except for the word "personnel number", you can do without the suffix "number". For example, we talk about a "vendor" and not a "vendor number".

Suffixes such as "key" and "indicator" should only be used if they are necessary to understand the field.

Texts in the Form of Questions

Texts containing questions should be positive.

Good:	<i>Do you want to save the data?</i>	
Bad:		<i>Don't you want to save the data?</i>

Shortened Sentences

If you use short formulations, use the sequence verb -> noun without the article.

Avoid nominalizations, that is, making verbs into nouns.

Exception: If several short phrases appear one after the other (for example, in a pull-down menu), the part containing the information - that is, the verb - is always written first.

Good:	<i>Enter account number</i>		
			<i>Display account number</i>
			<i>Delete account number</i>
Bad:		<i>Entry of the account number</i>	
			<i>Entry account number</i>
			<i>Account number display</i>
			<i>Account number delete</i>

Messages

Messages should be formulated as **precise** as possible. They should describe the current situation in **unambiguous** terms.

Formulations

They should not have a **negative** connotation and should not be in any way instructive. Meaningless or instructive words such as "may", "must", "should", "belong", "wrong", "incorrect", "incomplete", "not allowed", or the like are therefore to be avoided.

The **terms** used must be appropriate for the task of the user and be easy to understand. Messages which refer to technical modes of operation of the system are to be avoided.

The **syntactic structure** should be clear and concise and consist of complete sentences. You do not have to use articles, but at least an auxiliary verb must be present. Use the present tense whenever possible.

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Field Names \(Control Element\) \[Seite 324\]](#)
- [General Messages \[Seite 604\]](#)
- [Hyphenation \[Seite 620\]](#)
- [Message Types and Error Groups \[Seite 611\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Upper/Lowercase \[Seite 619\]](#)

Highlighting

You can achieve better legibility in the work area of a screen by highlighting important or structuring elements. There are several different types of highlighting.

Rules

Highlighting Types

From the ergonomic point of view, the possible types of highlighting can be divided into the following three groups:

1. **Tolerable** highlighting that is to be used very sparingly and only in the case of an emergency (see below) includes:
 - Lines as single lines, interrupted lines and double lines
 - Intensity
2. **Hardly tolerable** highlighting that is to be avoided, if possible, includes:
 - Semi-graphic symbols such as * ! #
3. **Ergonomically bad** highlighting that is **not** to be used on a screen includes:
 - Uppercase
 - Spaced text
 - Flashing
 - Color

Highlighting Usage

The highlighting in the first two groups is to be used only if all means of spatial and conceptual structuring have been exhausted.

As a rule, not more than 10 % of the screen should be extra emphasized. This portion is already exhausted by the lines for groups and lists or tables.

Therefore, try to avoid further highlighting under all circumstances.

Cross-References

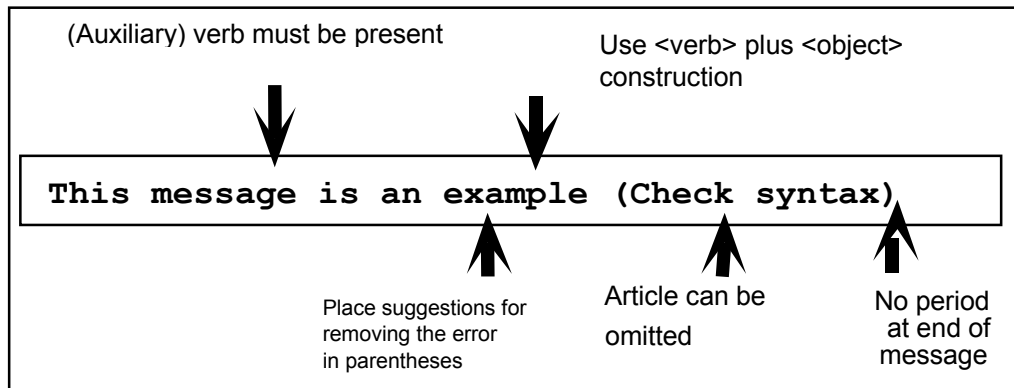
- [Fields - Positioning \[Seite 253\]](#)
- [Effects on Perception \[Seite 89\]](#)

General Messages

General Messages

If the system detects an error, it displays a message either in the status bar or in a dialog box, depending on the type of error. The message should state the problem and offer a solution to enable the user to proceed without outside assistance.

Illustration



The most important guidelines for writing messages

[Guidelines \[Seite 605\]](#)

[Examples of Good and Bad Error Messages \[Seite 607\]](#)

[Technical Notes \[Seite 609\]](#)

[Cross-References \[Seite 610\]](#)

Guidelines

Precise and Unambiguous

Be as **precise** as possible.

Messages should describe the situation which has occurred as **unambiguously** as possible.

Support

Offer the user **direct support** for the error recovery. Do not just state the problem, but rather indicate what needs to be done (see examples of Messages Prompting The User). If this is not possible, use a positive tone to explain the problem as precisely as possible. Additional support, if available, can be provided in parentheses.

Formulation

Adopt a positive tone, rather than being supercilious or condemning the user. Avoid the use of vague or judgemental words such as "may", "must", "should", "wrong", "insufficient", "incomplete", "not allowed", and so on.

User-Centered

Be specific and address the problem in the user's terms. Also avoid error messages which contain technical descriptions or explanations.

Syntactic Structure

Use clear and simple syntax and only write complete phrases (this ensures better understanding and also facilitates any translation process). Articles can be omitted; a verb (at least "is" or "are"), however, should always be present. Use the present tense, if possible.

Uppercase/Lowercase

Use normal upper and lower case conventions. Do not use upper case or spaced text for the purposes of emphasis. Exception: uppercase for report names, transaction codes, table names, file names.

Language

Be consistent in your use of language; do not mix different languages.

Punctuation

Do not conclude the message with a period or with an exclamation mark. Suggestions for the error correction can be given in parentheses. Always leave a space after a period and never before one.

Special Characters

Do not use special characters such as < >, as they will not necessarily be known to the user; value specifications such as % can be used, however. Messages are not to be introduced or concluded with asterisks (**). **Never** put field names, report names and table names in quotation marks.

Guidelines**Abbreviations**

Follow the guidelines for abbreviations (see [Abbreviations \[Seite 593\]](#)) and be consistent with the use of abbreviations.

Examples of Good and Bad Error Messages

The following sections list examples of bad and reformulated error messages (without comments) to illustrate the above-mentioned guidelines.

Messages Which can be Formulated More Specifically

Bad:	<i>Key & not in table &</i>
Good:	<i>Distribution channel & is not expected (Check your entry)</i>
Bad:	<i>The cursor is not placed on a line which can be selected</i>
Good:	<i>For the selection, position the cursor on a line of the table</i>
Bad:	<i>Release order specifications incomplete</i>
Good:	<i>Complete release order specifications by specifying order type, distribution channel</i>
Bad:	<i>List was not completely processed</i>
Good:	<i>List process cancelled at system limit (99 pages)</i>

Messages Formulated Less Ambiguously

Bad:	<i>The line & & & & is not in table 134</i>
Good:	<i>The combination of lines input is not valid</i>



It would be better to explain this message **in detail** in a dialog box. Or it is given in the short text as above, and the reference to the individual entry fields then appears in the long text.

Bad:	<i>Entries incomplete</i>
Good:	<i>Complete your entries</i>
Bad:	<i>No change</i>
Good:	<i>Entries were not copied, as they already exist</i>



It is important to formulate the message from the point of view of the user and **not** from the system's perspective.

Examples of Good and Bad Error Messages**User Prompt Messages**

Bad:	<i>Deletion flag incorrect, only 'J', 'S', ' ' allowed.</i>
Good:	<i>Enter 'J', 'S' or ' ' as deletion flags</i>
Bad:	<i>Postal code not entered or has incorrect length.</i>
Good:	<i>Enter postal code in defined length</i>
Bad:	<i>Percentage rate must be smaller than 100%</i>
Good:	<i>Enter percentage rate between 0 and 100</i>
Bad:	<i>No items selected</i>
Good:	<i>Select items in the list first</i>
Bad:	<i>Release date entered incorrectly</i>
Good:	<i>Enter the release date in the form MM.DD.YYYY</i>

Technical Notes

Messages like all message texts with language indicator and MESSAGE ID are maintained using transaction SE91. Relevant field contents can be inserted into the short text by using the placeholder "&". The current field contents are then appended to the MESSAGE statement with WITH in the order in which they are to be inserted into the short text. The MESSAGE ID must be specified at the beginning of the module pool in the PROGRAM statement:

```
PROGRAM <Name> MESSAGE ID <ID>.
```

By using WITH, you can append even more field contents to the MESSAGE statement than the number of placeholders contained in the short text, because the more detailed long text can require this. All field contents following WITH can be inserted into the long text in any sequence using "&V1&", "&V2&", "&V3&"... By means of the number, a reference is made to the corresponding field entry of the field contents listed behind WITH.

Messages in the ABAP are issued in the module pool by using the MESSAGE statement.

Cross-References

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Formulations \[Seite 601\]](#)
- [Message Types and Error Groups \[Seite 611\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Upper/Lowercase \[Seite 619\]](#)

Message Types and Error Groups

The messages in the R/3 System can be classified according to different types. The message type determines where the message is issued and how the system responds.

Messages which refer to errors, so-called error messages, can be grouped in terms of their contents on the basis of the events triggering them or the objects affected. Error messages are to state the problem and, if space permits, should offer suggestions for removing the error. In this way the user can proceed without outside help. Examples illustrate the individual error groups.

[Message Types \[Seite 612\]](#)

[Error Groups \[Seite 614\]](#)

[Technical Notes \[Seite 617\]](#)

[Cross-References \[Seite 618\]](#)

Message Types

Message Types



The message types have been defined by the programmers, not the Ergonomics Group.

S Message (Success)

S-type messages are displayed in the [Status Bar \[Seite 305\]](#) on the same or next screen. The message has no influence on the user's work. It only confirms the successful execution of system functions.

I Message (Information)

This message has a modal character - that is, the user must acknowledge the message. I messages are therefore displayed in a **modal dialog box**. The processing can be continued by pressing the `ENTER` key.

W Message (Warning)

So-called W messages interrupt the processing and allow the user to make corrections. For this reason, fields are enabled for input. W messages are displayed in the [Status Bar \[Seite 305\]](#) if the messages have been issued by a primary window. If a dialog box issues the message, the system displays the message in a separate dialog box.

With regard to the system response, warnings are similar to E messages. E messages, however, force the user to change the entry.

E Message (Error)

When the system detects an error, E messages should be used. Incorrectly completed fields must be ready for input. If consequential errors occur, we recommend to make not only the incorrectly filled field ready for input but also those fields whose entries have contributed to the error.

Depending on whether the E message was issued by a primary window or a dialog box, it is either displayed in the [Status Bar \[Seite 305\]](#) of the primary window or in a separate **dialog box**.

A Message (Abend)


A messages are intrinsically of a modal character, that is, a separate dialog is inserted into the current task. Therefore, they are displayed in **modal dialog boxes**.

A messages do not allow the user to make any further entries. The user can only acknowledge the message. The task is abruptly terminated and the system returns to a higher-level menu.

A messages should only be issued in extraordinary circumstances, for example, when a system-related error occurs or if the error can no longer be handled by the task. In all error situations handled by the task, however, it should be possible for the user to remain in his work environment. A messages should therefore be avoided, if possible.

Icons in Messages

If standard error messages from the Basis system are displayed in a dialog box, the system automatically displays the appropriate icon for the particular message type (E, W, I, or A). Since S messages always appear in the status bar, there is no icon for S messages.

	INFORMATION	I message		ERROR	E message
	WARNING	W message		CRITICAL	A message

Icons to be used in messages



In general, application developers do not have to take action in this respect.

Error Groups

Error Groups

Errors in Tables

Generally, the user is not authorized to make changes in a table. Therefore, the short text of the error messages which refer to tables should not contain a reference to a table. This information is contained in the long text, possibly together with the recommendation to contact the system administrator.

If the display of the message was initiated because a system table does not include an entry which it was expected to contain, the short text should not contain a reference to the table but at most a note on the entry:



Good: *E: Order type & is not valid (Check your entry)*

If the error is related to tables, which are only of importance for the SAP System itself or if the cause for the error cannot explicitly be assigned to a particular entry, the message should refer the user to the system administrator:



Good: *A: No further processing possible (Contact your system administrator)*

The long text should always contain a detailed error diagnosis which can also be of a more technical nature (which, however, is indicated as such).

If special reference is made to lines which are not contained in a table and if the table is important for the user, that is, if he has maintenance authorization for the table, use the following message pattern:



Good: *A: Enter country in country table & first*

Input Values are not within the Valid Value Range

The user has entered an entry which was not within the valid, usually numeric value range.



Good: *E: Enter posting period in allowed value range*

If the value range is known or predefined, the error message should ask the user to enter a value within a specified range:



Good: *E: Enter a posting period between 1 and 12*

Nonsensical Entries

In some situations, the user attempts to carry out an action which is meaningless from a business point of view (the system should be able to prevented these actions).



Good:	<i>E: Create bill of material first</i>
Good:	<i>E: Order type of outline agreement cannot be selected for &</i>
Good:	<i>E: Balance & is in the revaluation document of company code &</i>

System Error, System Limits are Exceeded

If a non-user error occurs in the system, the message should explain that a system-related problem has caused the error. It should also point out how the user can respond. Technical descriptions should be avoided.

For temporary problems, the following "good" example can be used:



Bad:	<i>I: Global lock table is full. Locking is currently notpossible</i>
Good:	<i>I: At present, the function cannot be carried out</i>

A detailed explanation of this message must be given in the long text.

If a more severe system error interrupts the user's interaction with the system, the following message can be displayed:



Good:	<i>A: No further processing is possible (Contact your system administrator)</i>
-------	---

A detailed explanation of this message must be given in the long text.

No Entry Made in a Certain Field

If particular entries are required to continue processing, the required specifications are, however, missing or incorrect:



Good:	<i>E: Enter material or reservation in case of a single entry</i>
Good:	<i>E: Enter the batch number for material &</i>
Good:	<i>E: Enter required-entry field material</i>

Control Error

If an operating error occurs, for example, the user chooses a function which requires a previous selection to have been made, the message should specify the initiated action and the error diagnosis, if possible:



Good:	<i>W: You have chosen Copy (Select an order first)</i>
Good:	<i>W: You have chosen Delete (Delete all items individually)</i>

Error Groups

Resource Conflicts Due to System Conditions

If records are locked, the user is missing the required authorization, etc.:



Good:	<i>I: You are not authorized for printer & (printer destination)</i>
Good:	<i>I: Vendor master record & is currently locked</i>

Invalid Result of an Action

If a calculation would lead to a meaningless or invalid result, not only the diagnosis should be given but also help in eliminating the error.



Good:	<i>E: The specified quantity is smaller than the delivered quantity (Repeat entry)</i>
Good:	<i>W: Today's purchase order & cannot be printed</i>

Success Messages

The system sends success messages to the user when it has performed certain tasks (e.g. particular important tasks).



Good:	<i>S: Differences to physical inventory document & have been posted with document &</i>
Good:	<i>S: New record with field & was added to list X</i>
Good:	<i>S: Physical inventory document with the generated number & has been posted</i>

Plausibility Error

The system, for early recognition of substantive errors, should check entries for their plausibility, if possible.



Good:	<i>W: Check the pricing</i>
-------	-----------------------------

Technical Notes

- In the flow logic of the screen, the required fields for E messages must be enabled for input by the FIELD statement in front of the module in question. After the user has corrected the entry, the module is processed again automatically from the beginning.
- Messages within modules are generally triggered at the event PROCESS AFTER INPUT. Only I messages and S messages are useful at the event PROCESS BEFORE OUTPUT.
- If several fields/modules are chained on the screen (CHAIN), the entire chain and therefore all modules are processed again.
- If a field is contained in several chains and an error occurs, all chains which are in front of the module reporting the error are checked for this field and processed again, if found.

Cross-References

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Formulations \[Seite 601\]](#)
- [General Messages \[Seite 604\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)
- [Status Bar \[Seite 305\]](#)
- [Upper/Lowercase \[Seite 619\]](#)

Upper/Lowercase

Guidelines

Upper/Lowercase

Use lower and uppercase letters. Do not use uppercase letters as design elements to highlight headers or other elements.



Good:	Software ergonomics is an	
		interdisciplinary science.
Bad:	SOFTWARE ERGONOMICS IS AN	
		INTERDISCIPLINARY SCIENCE.

Country-Specific Characters

Use country-specific characters, for example, the umlaut and ß in German.



Good:	Möhren ärgern Spaßvögel übrigens nie
Bad:	Moehren aergern Spassvoegel uebrigens nie

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [General Messages \[Seite 604\]](#)
- [Hyphenation \[Seite 620\]](#)
- [Overview \[Seite 469\]](#)
- [Prompts in Dialog Boxes \[Seite 50\]](#)

Hyphenation

Hyphenation

The following guidelines concern the hyphenation of words on the screen.

Guidelines

Do not hyphenate words on a screen, so that they extend over two lines.

Write words with a hyphen in the same line.



Good:	<i>Enter all your relevant data into this dialog box</i>	
		<i>The new SAP System is fairly good</i>
Bad:		<i>Enter all your relevant data into this dialog box</i>
		<i>The new SAP System is fairly good</i>



Contracting two word fragments separated by a hyphen and spread over two lines burdens the short-term memory of the user.

Exception

With long sections of continuous text (more than 50 characters), words can be hyphenated if this would otherwise result in large "gaps" on the right border.

Cross-References

- [Abbreviations \[Seite 593\]](#)
- [Formulations \[Seite 601\]](#)
- [Upper/Lowercase \[Seite 619\]](#)

Translation

This section is not meant to provide general guidelines for translating texts in the R/3 System into foreign languages. It is merely an attempt to group together the most important precautions that need to be taken to ensure that the R/3 System can be translated into foreign languages.

Basic Rule

Translation of texts from English into other languages may increase the length of the text by 30 percent or more. If you do not leave enough room on the screen, translators are required to abbreviate terms beyond recognition. This has a negative effect on the legibility of R/3 task screens.

As a rule of thumb, you should allow space for text expansion by about one quarter to one third during the translation.

[User Interface \[Seite 622\]](#)

[Screen Painter \[Seite 624\]](#)

[ABAP Text Pool \[Seite 625\]](#)

[Further Translation Problems \[Seite 627\]](#)

[Cross-References \[Seite 629\]](#)

User Interface

User Interface

Field names

For field names, the standard lengths provided by the system are to be used so that sufficient space is available for the translation. Sometimes, the second largest length can be used, but there should be at least two blanks.

Short Texts

The same applies to short texts as to field names. However, due to the greater length, the translators have more room to find an appropriate word for the available space.

Menu Texts

For menu texts, the system standard is 20 characters; it cannot be changed by developers. If the menu texts are very long and the entries no longer fit on one line of the menu bar, they have to be split.

With pull-down menu texts, there are no problems.

Pushbutton Texts

The same applies to pushbutton texts as to menu texts, that is, the system provides a maximum of 20 characters. For pushbuttons, however, you have to keep in mind some special features, although these do not affect the translation. The system displays as many pushbuttons as can be accommodated in the window depending on the text lengths of the individual pushbuttons. Thus, if the developer wants to make sure that a certain number of pushbuttons is displayed in the standard size window, he/she has to abbreviate the texts, if necessary. On the other hand, the number of pushbuttons increases when the window is enlarged, providing a sufficient number of pushbuttons is defined; correspondingly, if the window is reduced, fewer pushbuttons are displayed.

Since some general functions previously in the application toolbar have moved to the standard toolbar, you can now display more application-specific pushbuttons.

Function Texts

During the enhancement of a user interface, problems can occur when replacing a function text. If this change is used to eliminate a guideline violation and if it does not change the nature of the function, it is useful. Otherwise, please always create a new function with a new function code!

Reason: The foreign language does not necessarily have the same status as the current German version. This may lead to unwanted functions being called. In extreme cases, "Save" could be replaced by "Delete".

Fields in Entry Transactions

Entries in fields of regular entry transactions cannot be language-dependent, unless a new key is assigned. As long as no suited tools are available, an assignment of a new key from external to internal display and vice versa has to be programmed explicitly.

Principle: Do not use language-dependent keys without assigning a new key.

One-Character Fields

In the past, one-character fields were used to make single selections or multiple selections. In future, these fields should be replaced by graphic control elements.

More Than One Characteristic

One-character fields, which can have more than one characteristic and which are used for choosing alternatives, can only have the values 0, 1, 2..., 9 or A, B, C, D,...



Tables in which options are chosen.

Here, you have to check whether those fields can be converted into lists of radio buttons with mnemonic field names.

Exactly Two Characteristics

One-character fields, which have a toggle and which can have exactly two characteristics, can only have the values 'X' and blank (for example, "Overdelivery allowed", "selected", and so on). Here, one-character fields should also be converted into checkboxes (multiple selection) or radio buttons (single selection). This way, you can also make sure that the task receives the correct, that is, language-independent entries.

Screen Painter

Screen Painter

Name for Fields and Keywords

All fields and keywords on the screen must have a name.



When transporting screens, please check that all screen keywords with Data Dictionary reference also exist in the target system.



- Screen 1 gets the keywords from table TAB1 (data element keywords).
- Screen 1 is transported, but not TAB1.
- Screen 1 now contains the hard-coded German texts because no exchange with Data Dictionary texts can be made in the foreign language.

(If the screen transport was unsuccessful and there is no table field in the target system, both the table and the screen should be transported again during the post-editing so that the screen is generated in the target system and the keyword reference to the Data Dictionary is established.)

Screen Enhancement

During the screen enhancement, the field names for texts which are "hard-coded" on the screen used in the field list of the screen cannot be used again for other texts, if the original version is no longer needed.

Reason: The standard foreign language version does not always have the same status as the current original version.

ABAP Text Pool

Numbering

All texts that are defined in a program have to be stored in the text pool provided with a number. Otherwise, they cannot be translated.

Length

The length of the texts should be identified by an end mark (;:).

The end mark shows the translator the position at which the text must end.



Since the Basis department does not yet support the end mark, the texts may currently only be written by specifying the length.



WRITE: / (35) TEXT-001

Logically Related Texts

Texts that are logically related should also be stored as such in the text pool so that the relation can be recognized.



```
Delete A01 object.                ;;
A02 Do you really want to delete  ;;
object A03?                       ;;
Create object B01;;
B02 Do you really want to create  ;;
object B03?                       ;;
```

It is better to store texts twice (A02 = B02) than to define them as continuous texts that cannot be interpreted at runtime.

In this example, it is very dangerous to use the partial string A02 for both sentences. This results in complications in the foreign language.

Even if the texts can be translated, it is very difficult in almost all cases to create a syntactically correct sentence. The following example illustrates this.

English Example

Do you want to delete this object?	Do you want to create this object?
---------------------------------------	---------------------------------------

Here, the second part of the sentence is identical, compared to the first part in the original German sentence. If the partial string A02 was used for both sentences, the translator would be forced to stick to the German word order. It would be unreasonable to expect foreign language users to read such irregular (German) word orders in the foreign language:

Do you really want this object to delete?	Do you really want this object to create?
--	--

ABAP Text Pool

Compounds

Sentences that are put together by developers from several independent partial strings can never be translated correctly because, on the one hand, the translator does not know the context (at least, if the texts are scattered across the text pool) and, on the other hand, the word order cannot be changed.

Even in one-line sentences, the following text output could not be translated as the English translation shows:

```
Text 001 = 'Wollen Sie das Objekt'. 'Do you want object'  
Text 002 = 'löschen?'. 'to delete?'  
WRITE: Text 001, object name, text 002.
```

The following would be correct:

```
Text 001 = 'Do you want &1 to delete the object ?'.  
Help field = text 001.  
REPLACE '&1' WITH object name INTO help field.  
WRITE: Help field.
```

Text Numbers That are Available Again

Text numbers that are available for reuse should never be used for other texts during the program enhancement. Please always create a new text with a new number, because it cannot be guaranteed that the standard foreign language has the same status as the current German version.

Continuous Text

If possible, you should not store continuous text in the ABAP text pool.

In future, the display of continuous text within a list will be supported by the development environment. However, another clipboard will be used for this.

Data Element Keywords

The direct output of data element keywords (WRITE: DTEL-KEYWORD) within a list is currently not possible.

Further Translation Problems

ABAP Dictionary

Values defined in the ABAP Dictionary cannot be translated, only their short texts. You therefore cannot use mnemonic keys at all.

Help Views and Foreign Key Maintenance

Help views have to be maintained correctly. When doing this, please ensure that the type of relationship is also maintained correctly for the text tables, which must have the dependency factor 'TEXT'. Otherwise, the first found text (language-independent) is displayed when using the F4 help; currently, this is the German text because the language indicator 'D' has priority over all other language indicators (EFINSKOV).

T100 Messages and Titles in the Header Line

When translating messages and titles in header lines, assembling texts from different, possibly variable parts into a useful and linguistically perfect unit can be a problem.

Texts With More Than one Variable Symbol

Texts which contain more than one variable symbol can often not be translated meaningfully. Until a suitable Basis function is available, you should take this into account and use only one variable symbol for new messages. Already existing messages, however, should not be revised.

Numbered Variables

In future, variables will be numbered (&1, &2, &3, &4). However, even with numbering, a continuous text assembled dynamically from several variables will be either difficult to translate or may not allow any translation at all.

Continuous Text in Variables

Variables should never contain continuous texts.

Tables With Control Keys

Control keys that are stored in tables cannot be translated.

Keys of general importance, such as order type, document type, and so on, must therefore be language-independent. These fields have to be encrypted in numerical sequence, unless the application programs assign a new key externally or internally. In future, the Central Objects group will develop a conversion function for this purpose.

Included Documentation Elements

Within these elements, translation cannot be guaranteed, if you are using variables in IF constructs. Such constructs are to be avoided.

Forms

Leave Space!

During the output of texts on forms, you should provide enough space to avoid inconvenient abbreviations during the translation.

Further Translation Problems

We strongly recommend that the developer contacts the responsible translator in such cases.

Text Length

Output of texts that are shorter than the original field length (for example &TABCD-VTEXT(17)&) does not usually make sense with forms. During the translation, however, the length of the text output cannot be recognized.

Therefore, do not define text fields longer than necessary in the Data Dictionary, and write text fields in the defined length, if possible.

Changes

Since there is no proof of change that forms have been changed, you should at least document the changes in comment lines in the form itself (that is, changes since the last maintenance level upgrade). This helps the translators as well as the customers who use our forms or use them as a reference for their own forms.

SAPscript Standard Texts

Since SAPscript standard texts are neither delivered nor translated, they should not be included in forms.

Therefore, enter those texts directly in the form.

Relation to Business Objects

If possible, a form should belong to exactly one business object, for example the document type. It should not be used for different purposes by means of (unclear) dynamic programming, because otherwise it may not be possible to achieve a meaningful translation.

Tests

After the translation, developers should test the foreign language forms everywhere they are used. They should discuss any problems with the translator.

Country-Specific Forms

For country-specific forms, translators should consult the "Internationalization" Group.

Cross-References

- [Field Names \[Seite 324\]](#)
- [General Messages \[Seite 604\]](#)
- [Input/Output Fields \(Control Elements\) \[Seite 238\]](#)
- [Overview \[Seite 469\]](#)
- [Title Bar \[Seite 152\]](#)

International Terminology Lists

International Terminology Lists

[Begriffe der Menüleiste / menu bar / Barra de menús / Riga menù / Barre de menus \[Seite 636\]](#)

[Begriffe im Aktionsmenü unter <Objekt> / <Object> / <Objeto> / <Oggetto> / <Objet> \[Seite 637\]](#)

[Begriffe im Aktionsmenü unter Bearbeiten* / Edit / Editar / Elaborare / Traitement \[Seite 638\]](#)

[Begriffe im Aktionsmenü unter Springen / Goto / Pasar a / Saltar / Passer \[Seite 639\]](#)

[Begriffe im Dialogfenster / dialog box / / / fenêtre dialogue \[Seite 640\]](#)

[International Terminology Lists - Overview \[Seite 631\]](#)

[Standard Function Names \[Seite 633\]](#)

[Menu Bar Options \[Seite 635\]](#)

[Terms of the User Interface \[Seite 641\]](#)

International Terminology Lists - Overview

This section lists terms in different languages, primarily German and English, used in connection with the user interface. The terms are used in SAP GUI menus, for functions and user interface objects. They are also used in the documentation and the help system and maintained in the SAPterm glossary.

Terminology Lists

The "Standard Function Names" list contains the terms used for standard functions including function key settings. This is followed by a group of lists of menu terms (including function names and mnemonic access characters). The "User Interface Terms" list is a collection of various terms for user interface objects, standard functions and more general objects and actions among others. It is based on the German/English list of CUA Terms (Standard 1).

Languages

In addition to German and English terms, this section also lists Italian, Spanish, and some French terms. This list is relevant for rendering the system into other languages because translators often base their translations on the German and/or English terms.

The importance of the English terms for the international market does not need to be emphasized here. They are not only used in English-speaking countries, mainly in the USA

In addition to German and English terms, this section also lists Italian, Spanish, and some French terms. This list is relevant for rendering the system into other languages because translators often base their translations on the German and/or English terms.

The importance of the English terms for the international market does not need to be emphasized here. They are not only used in English-speaking countries, mainly in the U

Cross-Application Terminology Group

The German and English terms were discussed and agreed upon in the Cross-Application Terminology Group. The group will also discuss all new and additional terms. The SAP Terminology Group informs about new terms and also enters them into SAPterm. We are currently working on a procedure for maintaining the terminology of the other languages.

SAP Ergonomics Group

The SAP Ergonomics Group proposes new terms, for example, for new interface elements of the SAP GUI, and presents them to the Cross-Application Terminology Group. The Terminology Group discusses the proposals and agrees upon the terms to be used. Proposals of the Ergonomics Group are based on the Windows, Motif, CUA, and Apple user interface style guides but also on the documentation of widely used software products.

[Standard Function Names \[Seite 633\]](#)

[Menu Bar Options \[Seite 635\]](#)

Terms of the Menüleiste / menu bar / Barra de menús / Riga menù / Barre de menus

Terms of the pull-down menu <Objekt> / <Object> / <Objeto> / <Oggetto> / <Objet>

Terms of the pull-down menu Bearbeiten/ Edit / Tratar / Elaborare / Traitement

International Terminology Lists - Overview

Terms of the pull-down menu Springen / Goto / Pasar a / Saltar / Passer

Terms of the Dialogfenster / dialog box / ventana de diálogo/ _ / fenêtre dialogue

[Terms of the User Interface \[Seite 641\]](#)

Standard Function Names

F key	German	English	Spanish	Italian	French
	Anlegen	Create	Crear	Creare	Créer
	Ändern	Change	Modificar	Modificare	Modifier
	Anzeigen	Display	Visualizar	Visualizzare	Afficher
	Liste	List	Listar	Lista	Liste
	Ausführen	Execute	Ejecutar	Eseguire	Exécuter
F11	Sichern	Save	Grabar	Salvare	Sauvegarder
F11	Buchen	Post	Contabilizar	Registrare	Enregistrer
F16	Sichern ohne Prüfen	Save without check	Grabar sin verificar	Salvare senza verifica	Sauvegarder sans vérifier
F16	Merken	Hold	Retener	Memorizzare	Maintenir
	Generieren	Generate	Generar	Generare	Générer
	Prüfen	Check	Verificar	Verificare	Vérifier
F12	Abbrechen	Cancel	Cancelar	Interrompere	Interrompre
F3	Zurück	Back	Back	Indietro	Retour
F15	Beenden	Exit	Finalizar	Terminare	Terminer
	Ende <Anwendung>	Exit <application>	Fin <apl.>	Fine <appl.>	Fin <application>
	Schließen	Close	Cerrar	Chiudere	Fermer
F14	Löschen	Delete	Borrar	Cancellare	Supprimer
	Stornieren	Reverse bzw. Cancel <object> (context-specific)	Anular Stornare <oggetto> (context-specific)	Annullare	Annuler
	Widerrufen	Undo	Deshacer	Ripristinare	Défaire
	Rücknahme	Reset	Restablecer	Reset	Réinitialiser
F9	Markieren	Select	Marcar	Marcare	Marquer

Standard Function Names

F2	Auswählen	Choose	Seleccionar	Selezionare	Sélectionner
	Ausschneiden	Cut	Cortar	Cut	Couper
	Einfügen	Insert	Insertar	Inserire	Insérer
	Einsetzen	Paste	Pegar	Paste	Coller
	Verschieben	Move	Desplazar	Spostare	Déplacer
F13	Drucken	Print	Imprimir	Stampare	Imprimer
	Editor	Editor	Editor	Editor	Editeur
	Kurztext	Short text	Texto breve	Testo breve	Désignation
	Langtext	Long text	Texto explicativo	Testo esteso	Texte (descriptif)
	Kopieren	Copy	Copiar	Copiare	Copier
F16	Sortieren	Sort	Clasificar	Classificare	Trier
	Summieren	total (verb)	Sumar	Totalizzare	Totaliser
F1	Hilfe	Help	Ayuda	Help	Aide

Menu Bar Options

The following section lists menu bar options in several languages.

[Terms of the Menüleiste / menu bar / Barra de menús / Riga menù / Barre de menus \[Seite 636\]](#)

[Terms of the pull-down menu <Objekt> / <Object> / <Objeto> / <Oggetto> / <Objet> \[Seite 637\]](#)

[Terms of the pull-down menu Bearbeiten/ Edit / Tratar / Elaborare / Traitement \[Seite 638\]](#)

[Terms of the pull-down menu Springen / Goto / Pasar a / Saltar / Passer \[Seite 639\]](#)

[Terms of the Dialogfenster / dialog box / ventana de diálogo/ / fenêtre dialogue \[Seite 640\]](#)

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Documentation not Available in Release 4.6B

Terms of the User Interface

German	English	Spanish	Italian	French
Aktion	action	acción	Operazione	action
Aktionsmenü	pull-down menu	menú desplegable	Menù a tendina	menu déroulant
Aktionsmenü-eintrag	pull-down option*	opción del menú desplegable	Opzione menù a tendina	rubrique du menu déroulant
Aktives Fenster	active window	ventana activa	Finestra attiva	fenêtre active
Ankreuzfeld (Syn. Markierungsfeld, Markierfeld)	checkbox	campo de selección (sin. campo de marcado, campo de control)	Campo di selezione	zone à cocher
Anwendung	application	aplicación	Applicazione	application
Anwendungs-symbol	application icon	icono de aplicación	Icona di applicazione	icône d'application
Arbeitsbereich	work area	área de trabajo	Area operativa	zone de travail
Ausschneiden (Verb)	cut	cortar	Cut	couper
Auswahl	choice	selección	Selezione	sélection
nicht verfügbare A.	unavailable choice	selección no disponible	Selezione non disponibile	sélection non disponible
Standardauswahl	default choice	selección estándar	Selezione default	sélection standard
verfügbare A.	available choice	selección disponible	Selezione disponibile	sélection disponible
Auswählen (Verb)	choose	seleccionar	Selezionare	sélectionner
Auswahlknopf	radio button	botón de selección	Tasto di selezione	bouton de sélection
Befehl	command	comando	Comando	instruction
Befehlsfeld (Syn. Befehls-eingabefeld)	command field	campo de comando (sin. campo para entrada de comando)	Campo comando	zone d'instruction
Befehlsliste	command list	lista de comandos	Lista comandi	liste d'instructions
Befehlszeile	command line*	línea de comando	Riga comandi	ligne d'instruction

Terms of the User Interface

Bildschirm	screen	pantalla	Schermo	écran
Blätterleiste	scroll bar	barra de desplazamiento	Barra di scorrimento	barre de défilement
Blättermarke	slider box	desplazador	Ascensore	ascenseur
Blätterpfeile	scroll arrows	flechas de desplazamiento	Tasti di scorrimento	flèches de défilement
Cursor	cursor	cursor	Cursore	curseur
Dialogfenster	dialog box	ventana de diálogo	Box di dialogo	fenêtre dialogue
Direktanwahl	fastpath	acceso directo	Selezione diretta	sélection directe
Doppelklicken (Verb)	double-click	hacer doble clic	Fare doppio clic	double-cliquer
Doppelklick	double click	doble clic	Doppio clic	double-clic
Drucktaste	pushbutton(Short form: button)	pulsador	Pushbutton	bouton poussoir
Drucktastenleiste	application toolbar	barra de pulsadores	Stringa pushbuttons	barre de boutons poussoirs
Editieren (Verb)	edit	editar	Editare	éditer
Enter-Drucktaste	ENTER push-button	pulsador INTRO	Tasto d'invio	touche Enter
Einfügen (Verb)	insert	insertar	Inserire	insérer
Eingabefeld	entry field	campo de entrada	Campo input	zone de saisie ou zone d'entrée
Erweiterte Hilfe	extended help	ayuda ampliada	Help ampliato	aide étendue
F-Tasten anzeigen	display F keys	visualizar teclas-F	Visualizzare tasti F	afficher touches fonction
Fehlermeldung	error message	mensaje de error	Messaggio di errore	message d'erreur
Fenster	window	ventana	Finestra	fenêtre
Fenstertitel	window title	título de ventana	Titolo finestra	titre de fenêtre
Folgefunktion	follow-on	función siguiente function	Funzione successiva	fonction suivante
Funktionsleiste	toolbar	barra de funciones	Stringa funzione	barre de fonctions
Funktionstaste	function key	tecla de función	Tasto funzione	touche fonction
Funktionstastenbereich	function key area	área de teclas de	Area tasti funzione	--

Terms of the User Interface

Funktionstasten- menü (Unzul. Syn.: Mausmenü)	function key menu	menú de teclas	Menù tasti funzione	menu des touches fonction
Geteiltes Fenster	split window	ventana partida	Split window	fenêtre partagée
Gruppenrahmen	group box	marco de grupo	Group box	cadre
Gruppenüber- schrift	group heading	título de grupo	Intestazione gruppo	intitulé cadre
Hilfe	help	ayuda	Help	aide
Hilfe für F-Tasten	F keys help	ayuda para teclas-F	Help per tasti F	aide aux touches fonctions
Hilfe zur Hilfe	help for help	ayuda para ayuda	Ulteriore help	aide à l'aide
Hilfefenster	help window	ventana de ayuda	Finestra help	fenêtre d'aide
Inaktives Fenster	inactive window	ventana inactiva	Finestra inattiva	fenêtre inactive
Information	information message	mensaje informativo	Messaggio informativo	message d'information
Kannfeld	optional-entry field	campo opcional (sin. campo de entrada opcional)	Campo facoltativo	zone facultative
Klicken (Verb), Klick	click	hacer clic	Fare clic	cliquer, clic
Kopieren (Verb)	copy	copiar	Copiare	copier
Lerntext	tutorial help	guía de aprendizaje	Testo di apprendimento	didacticiel
Markieren	select	marcar	Marcare	marquer
Markierung	selection	marca de selección	Marcatura	marquage
Markierung	deselect	desmarcar	Eliminare	démarquer
aufheben (Syn. Entmarkieren, Markierung löschen)		(sin. borrar marcas)	marcatura	
Mauszeiger	pointer	puntero	Puntatore mouse	pointeur
Maximale Größe	maximum size	tamaño máximo	Grandezza massima	taille maximale
Maximierungs- symbol (Syn. Vergrößerungs- symbol)	maximize button	icono de maximización (sin. botón maximizar)	Icona di massimizzazione	case d'agrandissemen t

Terms of the User Interface

Menü	menu	menú	Menù	menu
Menüleiste	menu bar*	barra de menús	Stringa menù	barre de menus
Menüleisten- eintrag	menu bar option*	opción en barra de menús	Opzione stringa menù	rubrique barre de menus
Minimierungs- symbol (Syn. Verkleinerungs- symbol)	minimize button	icono de minimización (sin. botón minimizar)	Icona di minimizzazione	case de réduction
Mußfeld	required-entry field (Syn. mandatory- entry field)	campo obligatorio (sin. campo de entrada obligatoria)	Campo obbligatorio	zone obligatoire
Nachricht	message	mensaje	Messaggio	message
Nachrichtenfenst er	message pop- up	ventana de mensaje	Finestra messaggio	fenêtre de message
Nachrichtenzeile	message area	línea de mensaje	Riga messaggio	ligne de message
Primärfenster	primary window	ventana primaria	Finestra principale	fenêtre principale
Rahmen	border	marco	Cornice	cadre
Sanduhr	hourglass pointer	reloj de arena	Clessidra	sablier
Schließen (Verb)	close	cerrar	Chiudere	fermer
Sekundärfenster	secondary window	ventana secundaria	Finestra secondaria	fenêtre secondaire
Spaltenüberschri ft	column heading	título de columna	Intestazione colonna	intitulé de colonne
Statusleiste	status bar	barra de status	Stringa stati	barre d'état
Symbol (Ikone)	icon	icono	Icona	icône
Symbol (grafische Schaltfläche)	Pushbutton (w. graphic) (short form: button)	pulsador	Pushbutton	bouton poussoir
Symbol (allg.)	symbol	símbolo	Icona	symbole
Symbolleiste	standard toolbar	barra de símbolos	Stringa icone	barre d'icônes
Systemmenü	system menu	menú del sistema	Menù sistema	menu système
Systemmenü-	system menu	icono del menú del	Icona menù	case de menu

Terms of the User Interface

symbol	button	sistema	sistema	système
--------	--------	---------	---------	---------

Tabulator	tab	tabulador	Tabulatore	tabulateur
Tastencode	accelerator	código de teclas	Codice tasti	raccourci
Titelleiste	title bar	barra de títulos	Stringa titoli	barre de titre
Übernehmen	adopt, copy (je nach Kontext)	adoptar	Rilevare	reprendre
Verschieben	move	mover	Spostare	déplacer
Vorschlagswert	default value	valor propuesto	Valore default	valeur par défaut
Vorschlagswert des Feldes	default field value	valor propuesto para campo	Valore default del campo	valeur par défaut de la zone
Warnung	warning message	mensaje de advertencia	Messaggio di avvertimento	message d'avertissement
Widerrufen (Verb)	undo	deshacer	Ripristinare	défaire
Wiederherstellen (Verb)	restore	restaurar	Ripristino	restaurer
Wiederherstellungs-symbol	restore button	icono de restauración	Icona di ripristino	case de restauration
Ziehen (Verb)	drag	arrastrar	Trascinare	glisser
Zwischenablage	clipboard	portapapeles	Clipboard	corbeille

*Notes

"Pull-down option": In long texts, the term "Menu option" is often used for this term.

"Command line": This term is only valid until Release 2.0.

"Menu bar": The term "Action bar" should no longer be used.

"Menu bar option": In long texts, the term "Menu option" is often used for this term.

"Extras": It can cover both "Additional functions" as well as "Additional data".

Overview of Standard Functions

Overview of Standard Functions

This appendix lists in alphabetical order all R/3 standard functions including specifications for






- menu texts, pushbutton texts and abbreviations (if defined)
- pull-down menus
- function keys including the group of restrictions
- pushbuttons (specifying the pushbutton block)
- mnemonic access character

[Functions \[Seite 647\]](#)

[Legend \[Seite 656\]](#)

Functions

Menu text Pushbutton text Abbreviation	Menu	Function key	Pushbutton group	Access character	Icon	
<u>B</u>ack						
Back	Goto	F3 (1)	Navigation	B		ST
Back		perm. active				


<u>C</u>ancel						
Cancel	Edit	F12 (1) and ESC	Navigation	A		ST
Cancel		perm. active				DB
Canc.						
<u>C</u>hange						
Change	<object>	---	---	C	 AT	
Change	3rd line					
Chnge						
<u>C</u>heck						
Check	<object>	can be defined	self-defined	-	 AT	
Check	Function grp	freely				
Chck	"Other"					
<u>C</u>hoose						
Choose	Edit	F2 (2)	Standard	H	 AT	
Choose	Double click					
Choose						
<u>C</u>ollective entry						
Collective entry	<object>	---	---	-		
---	2nd line					

Functions

Collective processing						
--- 2. line						

Collective processing	<object>	---	---	-		
Continue or OK						
Continue or OK	---	ENTER	OK	-	 ST	
---					 DB	

Copy						
Copy	Edit	can be defined	self-defined	C	 AT	
Copy		freely			 ST	
Copy					 obj.	
Create						
Create	<object>	---	---	E	 AT	
Create	2nd line					
Create						
Cut						
Cut	Edit	can be defined	self-defined	T	 AT	
Cut		freely			 ST	


Delete						
Delete	O: <object>	F14 (3)	Standard	D	 AT	
Delete	OC: Edit				 AT	
Del.					line	
Deselect all						
Deselect all	Edit	can be defined	self-defined	D	 AT	
---		freely				


Functions


Display						
Display	<object>	---	---	D	 AT	
Display	4th line					
Disp.						
Display <object>						
Display <object>	<object>	---	---	-		
---	following Save					
---	or Post					
Edit						
Edit	<object>	---	---	E		
---	2nd line					

Editor						
Editor	Goto	F9 (3) P	Standard	E	 AT	
Editor						
Edit.						
Enter						
Enter	<object>	---	---	-		
---	2. line					
Execute						
Execute	<object>	can be defined	self-defined	X	 AT	
Execute		freely				
Exec.						
Exit (end of <application area>)						
Exit	<object>	F15 (1)	Navigation	X	 ST	
---	last line	perm. active				



Functions

Find						
Find	<object>	CTRL + F	Standard	F		ST*
Find	Edit					

Find next						
Find next	<object>	CTRL + G	Standard			ST*
Find next	Edit					



First page						
First page	---	CTRL + PAGEUP (1)	---	-		ST*
First page						

First screen						
First screen	Goto	can be defined	self-defined	-		
---		freely				
		not F21!				



Generate						
Generate	<object>	can be defined	self-defined	T		AT
Generate		freely				
Gen.						
Help						
Help	---	F1 (1)	---	-		ST
Help		perm. active				

Hold						
Hold	<object>	F16 (3) P	Standard	O		
Hold						
Hold						


Functions

<u>I</u>nsert <object>						
Insert <object>	Edit	can be defined	self-defined	I		AT
Insert <object>		freely			line	
Ins.						
<u>L</u>ast page						
Last page	---	CTRL +	---	-		ST
Last page		PAGEDOWN (1)				

<u>L</u>ast screen						
Last screen	---	can be defined	self-defined	-		
Last screen		freely				


<u>L</u>ist						
List	Goto	can be defined	self-defined	-	 AT	
List		freely				
List						
<u>L</u>ong text						
Long text	Goto	F9 (3) P	Standard	L		AT
Long text						


<u>M</u>enu bar						
Menu bar	---	F10 (1)	---	-		
Menu bar		perm. active				


<u>M</u>ove						
Move	Edit	can be defined	self-defined	M		AT
Move		freely				

Functions

<u>N</u>ew lines						
New lines	Edit	can be defined	self-defined	N		
New lines		freely				

<u>N</u>ext <object component>						
Next <object comp.>	Goto	F19 (2)	Navigation	N		AT
<object>+						



<u>N</u>ext page						
Next page	---	PAGERDOWN N (1)	---	-		ST*
Next p.						


<u>N</u>ext screen						
Next screen	Goto	can be defined	self-defined	-		
Screen+		freely				
---		not F23!				
<u>O</u>ther <object>						
Other <object>	<object>	F17 (2)	Navigation	O		AT
Other <object>		perm. active				


<u>O</u>ther <object component>						
Other <object comp.>	Goto	F20 (3)	Navigation	O		


<u>O</u>pen						
Open	Edit	F17 (2) nP	self-defined	O		
Open						



Functions

Paste						
Paste	Edit	can be defined	self-defined	P		AT
---		freely				ST


Possible entries						
Possible entries	---	F4 (1)	---	-		(ST)
PossEntr		perm. active				



Post						
Post	<object>	F11 (1)	1. pushbutton	?		
Post						
Post						
Previous <object component>						
Previous <object comp.>	Goto	F18 (2)	Navigation	P		AT
<object>-						


Previous page						
Previous page	---	PAGEUP (1)	---	-		ST*
Previous p.						

Previous screen						
Previous screen	Goto	can be defined	self-defined	-		
Screen-		freely				
---		not F22!				
Print						
Print	O: <object>	CTRL + P (3)	Standard	P		ST
Print	OC: Edit					DB
Prt						

Functions

Reset						
Reset	Edit	can be defined	self-defined	-		
Reset		freely				
Reset						
Reverse						
Reverse	<object>	can be defined	self-defined	-		
Reverse		freely				
Rev.						
Save						
Save	<object>	F11	1. pushbutton	S		ST
Save		and CTRL + S (1)				
Save						
Save without check						
Save without check	<object>	F16 (3) P	Standard	-		
Save without check						





Select/deselect						
Select/deselect	---	F9 (3) nP	Standard	-		AT
Select/deselect						ST
Sel.						
Select all						
Select all	Edit	can be defined	self-defined	S		AT
---		freely				

Select block						
Select block	Edit	can be defined	self-defined	B *		AT
---		freely				

Functions

Single entry						
Single entry	<object>	---	---	-		
---	2. line					

Single processing						
Single processing	<object>	---	---	-		
---	2. line					

Sort						
Sort	Edit	F16 (3) nP	Standard	S		AT
Sort						AT
Sort						
Total						
Total	Edit	can be defined	self-defined	-		AT
Total		freely				
Tot.						
Undo						
Undo	Edit	can be defined	self-defined	U		AT
Undo		freely				

Legend

Legend

Function Keys

- P has priority if assigned twice
- nP has no priority if assigned twice. Choose any function key.
- AT Application toolbar (icon); may usually also be used in the work area
- ST Standard toolbar (icon, pushbutton group)
- ST* Standard toolbar (icon, pushbutton group); may also appear in the work area
- WA Work area (icon, pushbutton group)
- DB Dialog box (icon, pushbutton group)
1. *Function keys group 1* = Keys standard at SAP: functions are linked to the function keys of this group and should be permanently available; these keys may **never** be used for other functions even if the related function is not used or inactive in the task.
 2. *Function keys group 2* = Mandatory function keys, in case the function is activated sometimes: these function keys belong to functions which are not always available. If such a function is used at least once in a task, the defined function key has to be used for this function; otherwise, it can also be used for other functions.
 3. *Function keys group 3* = Recommended function keys, in case the function is activated sometimes: function keys which should be assigned to the defined functions (recommendation), if they are active; otherwise, they can be used for other functions.



The functions of group 1 and 2 can **not be active without the respective function keys**. If the functions of group 2 are not used, the corresponding function keys can be used for other functions.

There may be other functions **with or without** the respective function keys (group 3 and 4).

Finally, a number of functions may not be assigned to **any** function key or included in the application toolbar or do not have a pull-down menu; this is indicated by "----".

R/3 Icons and Symbols

For the R/3 System, a number of icons and symbols has been defined. This section contains an overview of these.

[Icons \[Seite 658\]](#)

[Symbols \[Seite 672\]](#)

Icons

Icons

This section presents icons according to their use, which means that some icons are listed several times. In the R/3 System, you can display an alphabetical list of icons with transaction ICONS.

[Icons on Pushbuttons in the Standard Toolbar \[Seite 659\]](#)

[Icons on Pushbuttons in the Application Toolbar \[Seite 660\]](#)

[Pushbuttons with Icons on Data Screens \[Seite 662\]](#)

[Pushbuttons with Icons on Initial Screens \(Tools\) \[Seite 664\]](#)

[Application-specific Functions \(ABAP/4 Workbench\) \[Seite 665\]](#)

[Special Icons for Networks \[Seite 666\]](#)

[Report Selection Screens \[Seite 667\]](#)
















[Specific Icons on Pushbuttons in Dialog Boxes \[Seite 668\]](#)

[Icons for Messages and System Prompts \[Seite 669\]](#)

[Icons as Field Names \[Seite 670\]](#)

[Icons as Status Displays \[Seite 671\]](#)

Icons on Pushbuttons in the Standard Toolbar

	SAVE	<i>Save</i>		FIRST_PAGE	<i>First Page</i>
	BACK	<i>Back</i>		PREVIOUS_PAGE	<i>Previous Page</i>
	EXIT	<i>Exit</i>		NEXT_PAGE	<i>Next Page</i>
	CANCEL	<i>Cancel</i>		LAST_PAGE	<i>Last Page</i>
	SYSTEM_PRINT	<i>Print</i>		MODUS_CREATE	<i>Create Session</i>
	SEARCH	<i>Find</i>		SHORTCUT	<i>Create Shortcut</i>
	SEARCH_NEXT	<i>Find Next</i>		HELP	<i>Help</i>
				SYSTEM_SETTINGS	<i>Layout Menu</i>

Icons for standard toolbar functions

Icons on Pushbuttons in the Application Toolbar

Icons on Pushbuttons in the Application Toolbar

Icons without text label

	EXECUTE_OBJECT	<i>Execute</i>		TOTAL_LEFT	<i>First item/position</i>
	REPLACE_OBJECT	<i>Replace</i>		COLUMN_LEFT	<i>Previous item/Position</i>
	COPY_OBJECT	<i>Copy</i>		COLUMN_RIGHT	<i>Next item/position</i>
	DELETE	<i>Delete</i>		TOTAL_RIGHT	<i>Last item/position</i>
	SORT_UP	<i>Sort (up)</i>		SORT_DOWN	<i>Sort (down)</i>
	SEARCH	<i>Find</i>		SEARCH_NEXT	<i>Find next</i>
	SUM	<i>Total</i>		OTHER_OBJECT	<i>Other object</i>
	PREVIOUS_OBJECT	<i>Previous object component, Previous object</i>		NEXT_OBJECT	<i>Next object component, Next object</i>
	COMPARE	<i>Compare</i>		FILTER	<i>Filter</i>
	RENAME	<i>Rename</i>		TRANSPORT	<i>Tranport</i>
	EXPORT	<i>Export, Download</i>		IMPORT	<i>Import, Upload</i>
	SYSTEM_CUT	<i>Cut (central)</i>		SYSTEM_COPY	<i>Copy (central)</i>
	SYSTEM_PASTE	<i>Paste (central)</i>		SYSTEM_MARK	<i>Select mode, Select/Deselect</i>
	SYSTEM_UNDO	<i>Undo</i>		SYSTEM_REDO	<i>Redo</i>



Icons for application toolbar functions

	NEXT_VALUE	<i>Previous item, next value</i>		PREVIOUS_VALUE	<i>Next item, Previous value</i>
--	------------	----------------------------------	--	----------------	----------------------------------

Icons used in dialog boxes with reference to list items (spin button icons)









	EXPAND	<i>Expand</i>		COLLAPSE	<i>Collapse</i>
--	--------	---------------	--	----------	-----------------

Icons on Pushbuttons in the Application Toolbar

	NEXT_ HIER_LE VEL	<i>Next hierarchy level</i>		PREVIOUS_ HIER_LEVEL	<i>Previous hierarchy level</i>
---	-------------------------	-----------------------------	---	-------------------------	---------------------------------

Icons for functions on hierarchies

Icons with text label







	REFRESH	<i>Refresh</i>		HEADER	<i>Header data, Basic data</i>
	SKIP	<i>Skip</i>		OVERVIEW	<i>Overview data</i>
	CONVERT	<i>Convert</i>		SELECT_ DETAIL	<i>Choose (detail)</i>
	RETRIEVE	<i>Retrieve</i>		LIST	<i>List</i>

Icons for application toolbar functions that are usually accompanied by text





Pushbuttons with Icons on Data Screens

Pushbuttons with Icons on Data Screens

Horizontal and Vertical Scrolling in Tables



<i>Total left</i>	<i>Page left</i>	<i>Column left</i>	<i>Column right</i>	<i>Page right</i>	<i>Total right</i>
					
TOTAL_ LEFT	PAGE_ LEFT	COLUMN_ LEFT	COLUMN_ RIGHT	PAGE_ RIGHT	TOTAL_ RIGHT

Icons for horizontal scroll functions (use the arrangement show)

<i>First page (F21)</i>	<i>Previous page (F22)</i>	<i>Next page (F23)</i>	<i>Last page (F24)</i>
			
FIRST_PAGE	PREVIOUS_PAGE	NEXT_PAGE	LAST_PAGE







Icons for vertical scroll functions (arranged as shown or vertically from left to right)

Spin Buttons

	NEXT_VALUE	<i>Previous item, next value</i>		PREVIOUS_VALUE	<i>Next item, Previous value</i>
---	------------	----------------------------------	---	----------------	----------------------------------

Icons used for spin buttons











Word Processing and Annotation Functions

	CHANGE_TEXT	<i>(Change) text</i>		CREATE_TEXT	<i>Create text</i>
				DISPLAY_TEXT	<i>Display text</i>
	ANNOTATION	<i>(Change) annotation</i>		CREATE_NOTE	<i>Create note</i>
				DISPLAY_NOTE	<i>Display note</i>

Icons for text/word processing and annotation functions

















Other Functions

Pushbuttons with Icons on Data Screens

	CALCULATION	<i>Spreadsheet, Calculation</i>		GRAPHICS	<i>Graphics</i>
	SHORT_MESSAGE	<i>Short message</i>		ADDRESS	<i>Address</i>
	INFORMATION	<i>Information</i>		SET_STATE	<i>Set state</i>
	VARIANTS	<i>Variants</i>		TOOLS	<i>Tools</i>
	EXPORT	<i>Export, Download</i>		IMPORT	<i>Import, Upload</i>

Icons for other functions











Functions for Tables

	MOVE	<i>Move</i>		REPLACE	<i>Replace</i>
	INSERT_ROW	<i>Insert row</i>		DELETE_ROW	<i>Delete row</i>
	SEARCH	<i>Search</i>		SEARCH_NEXT	<i>Search next</i>
	SORT_UP	<i>Sort up</i>		SORT_DOWN	<i>Sort down</i>
	SUM	<i>Totals</i>		SELECT_DETAIL	<i>Select detail</i>
	EXPAND	<i>Expand</i>		POSITION	<i>Position</i>
	COLLAPSE	<i>Collapse</i>		SELECT_BLOCK	<i>Select Block</i>
	SELECT_ALL	<i>Select all</i>		DESELECT_ALL	<i>Deselect all</i>

Icons for tables













Pushbuttons with Icons on Initial Screens (Tools)

Pushbuttons with Icons on Initial Screens (Tools)

	CREATE	<i>Create</i>		REPLACE	<i>Replace</i>
	CHANGE	<i>Change</i>		COPY_OBJECT	<i>Copy<object></i>
	DISPLAY	<i>Display</i>		DELETE	<i>Delete</i>
	TOGGLE_ DISPLAY_ CHANGE	<i>Display <-> Change</i>		EXECUTE_ OBJECT	<i>Execute</i>
	SEARCH	<i>Find</i>		SEARCH_NEXT	<i>Find next</i>

Icons for generic functions on initial screens








Application-specific Functions (ABAP/4 Workbench)

	ACTIVATE	<i>Activate</i>		ICON_LIST	<i>Icon list</i>
	GENERATE	<i>Generate</i>		STACK	<i>Stack</i>
	CHECK	<i>Check</i>		REFERENCE_ LIST	<i>Reference list</i>
	TEST	<i>Test</i>		TRANSPORT	<i>Transport</i>
	COMPARE	<i>Compare</i>		TOOLS	<i>Tools</i>
	RENAME	<i>Rename</i>		BREAKPOINT	<i>Breakpoint</i>

Special functions (Workbench)





















Special Icons for Networks

Special Icons for Networks

	PREVIOUS_NODE	<i>Previous node</i>		NEXT_NODE	<i>Next node</i>
	PREVIOUS_HIER_LEVEL	<i>Previous hierarchy level</i>		NEXT_HIER_LEVEL	<i>Next hierarchy level</i>
	TREE	<i>Tree</i>		INSERT_RELATION	<i>Insert relation</i>
	FINAL_DATE	<i>Final date</i>			

Special icons for networks




Report Selection Screens

	DISPLAY_ MORE	<i>Display more</i>		ENTER_MORE	<i>Enter more</i>
	NOT_EQUAL_ GREEN	<i>Not equal</i>		NOT_EQUAL_RED	<i>Not equal</i>
	GREATER_ GREEN	<i>Greater</i>		GREATER_RED	<i>Greater</i>
	LESS_GREEN	<i>Less</i>		LESS_RED	<i>Less</i>
	GREATER_ EQUAL_GREEN	<i>Greater or equal</i>		GREATER_ EQUAL_RED	<i>Greater or equal</i>
	LESS_EQUAL_ GREEN	<i>Less or equal</i>		LESS_EQUAL_ RED	<i>Less or equal</i>
	INTERVAL_ EXCLUDE_ GREEN	<i>Include interval</i>		INTERVAL_ INCLUDE_RED	<i>Include interval</i>
	INTERVAL_ EXCLUDE_ GREEN	<i>Exclude interval</i>		INTERVAL_ EXCLUDE_RED	<i>Exclude interval</i>
	PATTERN_ INCLUDE_ GREEN	<i>Include pattern</i>		PATTERN_ INCLUDE_RED	<i>Include pattern</i>
	PATTERN_ EXCLUDE_ GREEN	<i>Exclude pattern</i>		PATTERN_ EXCLUDE_RED	<i>Exclude pattern</i>

Icons for ABAP/4 report selection screens (green = select, red = exclude from selection)





Specific Icons on Pushbuttons in Dialog Boxes

Specific Icons on Pushbuttons in Dialog Boxes

	OKAY	<i>Execute, Continue</i>
	CANCEL	<i>Cancel</i>
	PRINT	<i>Print</i>

Specific icons for functions in dialog boxes

Icons for Messages and System Prompts

	INFORMATION	I message		ERROR	E message
	WARNING	W message		CRITICAL	A message

Icons to be used in messages







	WARNING	Loss of data		QUESTION	All others
---	---------	--------------	---	----------	------------

Icons to be used in safety or confirmation prompts

Icons as Field Names


Icons as Field Names

[Textfelder \[Seite 324\]](#)







	PHONE	Phone number		TIME	Time
	FAX	Fax number		DATE	Date
	Mail	Send mail		ALARM	Alarm

Icons that may be used as field names or as part of field names







Icons as Status Displays

	CHECKED	Checked			
---	---------	---------	--	--	--

Icons as status display: implicitly two-state

	CHECKED	Yes		INCOMPLETE	No
	LOCKED	Locked		UNLOCKED	Unlocked
	POSITIVE	Positive		NEGATIVE	Negative

Icons as status display: two-state (icons in rows belong together)

	CHECKED	Checked		GREEN_LIGHT	Positive
	INCOMPLETE	Incomplete		YELLOW_LIGHT	Neutral
	FAILURE	Failure		RED_LIGHT	Negative

Icons as status display: three-state (icons in columns belong together)

Symbols

Symbols

The symbols for lists are organized according to their main usage. A complete list of the symbol character set appears at the end of this chapter. For a list of all symbols with recommendations for their use, please refer to transaction LIBS.

[Symbols for Reporting Tasks \[Seite 673\]](#)

[Symbols for Office Applications \[Seite 674\]](#)

[Symbols for Use as Status Displays \[Seite 675\]](#)

[Symbols showing the Status of Objects, Numbers, Data Columns etc. \[Seite 676\]](#)













[Symbols for Navigation Functions \[Seite 677\]](#)

[Overview of the Symbol Character Font SAPDings \[Seite 678\]](#)



The ABAP/4 names of the symbols start with the prefix “SYM_”. This prefix is omitted in the following lists!


























Symbols for Reporting Tasks

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
	ASTERISK	Totals (at different levels)		DIAMOND	Subtotals
	DOCUMENT	Document		CHECKED_DOCUMENT	Checked or changed document
	DOCUMENTS	Dokuments, copy		NOTE	Note, annotation
	FOLDER	Folder		OPEN_FOLDER	Open folder
	CHECK_MARK	Status indicator		BOLD_X	Wrong or invalid value
	FILLED_CIRCLE	Mark, status indicator "positive"		CIRCLE	Mark, status indicator "negative"

Symbols for use in reporting tasks

Symbols for Office Applications

Symbols for Office Applications

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
	DOCUMENT	Document		CHECKED_DOCUMENT	Checked or edited document
	DOCUMENTS	Documents, copy		NOTE	Note, annotation
	FOLDER	Folder, filing		OPEN_FOLDER	Open folder
	DOCUMENT & GLASSES	Read, display document (2 symbols)		DOCUMENT & PENCIL	Write, create document (2 symbols)
	GLASSES	Read-only		PENCIL	Read-and-Write
	CHECK_MARK	Status: read, sent, etc.		DELETE	Deletion flag, marked for deletion
	UNLOCKED	Accessible, unlocked, etc		LOCKED	Protected, locked, etc.
	RIGHT_TRIANGLE	Marking		PHONE	Phone
	PRINTER	Printer, print		FAX	Fax machine
	LEFT_HAND	Hint		RIGHT_HAND	Hint
	WORK_ITEM	Workflow item		EXECUTABLE	Executable
	FLASH	Urgent, pressing		CAUTION	Caution, warning, deadline
	DOCUMENT & FLASH	Express document (2 symbols)			

Symbols for use in office applications

Symbols for Use as Status Displays

✓ CHECK_MARK	Edited, read, available (present), executable etc.	Empty	Unedited, unread, unavailable (absent), not executable etc.
-----------------	--	-------	---

Symbol as status display: implicitly two-state

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
✓	CHECK_MARK	O.k.	✗	BOLD_X	Not o.k.
●	FILLED_CIRCLE	Available	○	CIRCLE	Not available
🔒	LOCKED	Locked, protected	🔓	UNLOCKED	Unlocked, accessible











Symbol as status display: two-state (symbols in a row belong together)

● FILLED_CIRCLE	Finished, whole, full	◐ HALF_FILLED_CIRCLE	In process, partial, half	○ CIRCLE	Unprocessed, nothing, empty
--------------------	-----------------------------	-------------------------	---------------------------------	-------------	-----------------------------------

Symbol as status display: three-state

Symbols showing the Status of Objects, Numbers, Data Columns etc.

Symbols showing the Status of Objects, Numbers, Data Columns etc.

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
	GLASSES	Read-only		PENCIL	Read-and-Write
	LOCKED	Protected, locked		UNLOCKED	Accessible, unlocked
	BOLD_X	Invalid, error		FILLED CIRCLE & CIRCLE	Division by zero
	DELETE	Marked for deletion		CUMULATED	Column summed up, accumulated
	SORTED_UP	Column sorted up		SORTED_DOWN	Column sorted down

Symbols showing the status of objects, numbers, data columns etc.

Symbols for Navigation Functions

	ABAP/4 name	Meaning		ABAP/4 name	Meaning
◀	LEFT_TRIANGLE	Move back one horizontal unit	▶	RIGHT_TRIANGLE	Move forward one horizontal unit
▲	UP_TRIANGLE	Move up one vertical unit	▼	DOWN_TRIANGLE	Move down one vertical unit

Symbols for navigation functions

Overview of the Symbol Character Font SAPDings

Overview of the Symbol Character Font SAPDings

SAPDings 24 







Overview of the symbol character font SAPDings

Overview of Menu Bars

The following overview describes the basic hierarchy of the menu bars at the three levels of the SAP System, the main menu level, the application level, and the task level. The task level itself can have a hierarchical structure, too. Nevertheless, the menu bar should be structured uniformly within a task to ensure consistency. Functions that are not used should therefore not be removed from the menu bar and the pull-down menus but only be inactivated (menu item should be dimmed).

[General Remarks on Menu Bars \[Seite 680\]](#)

[Menu Bar at Task Level \[Seite 681\]](#)

[<Object> \[Seite 682\]](#)

[Edit \[Seite 683\]](#)

[Other Menus \[Seite 684\]](#)

General Remarks on Menu Bars

General Remarks on Menu Bars

System-wide Menus

The *System* and *Help* menus are always available in the SAP System. They are appended automatically to the right of the individual menus at the respective level. Therefore, they are not discussed further here.

Main Menu Level

In the menu bar at the Main Menu Level, the R/3 System displays all applications available. The following application areas are currently available: **Office, Logistics, Accounting, Human resources, Tools**.

Application Level

At the Application Level, **object classes** or **tasks = transactions** of an application area are listed in the menu bar. The user chooses an object class and a respective action and then starts a task. For grouping together object classes and transactions, please refer to the Menu Bar at Application Level section.

Task Level

At the Task Level, the user edits data. The structure of the menu bar and the corresponding pull-down menus are largely standardized. The following section gives an overview of the individual menu bar options in the defined sequence from left to right, together with a short definition and obligatory or typical pull-down menus.

Menu Bar at Task Level

Legend

Bold *menu bar option*: obligatory or optional

Normal *menu bar option*: to be used only if required

Bold *pull-down option*: obligatory

Normal *pull-down option*: optional

--- = here you can insert hyphens.

[<Object> \[Seite 682\]](#)

[Edit \[Seite 683\]](#)

[Other Menus \[Seite 684\]](#)

<Object>

<Object>

Actions which affect the entire object; they receive the name of the object that has been selected at the next higher level - **obligatory**

1. Initial screens:

- Actions for choosing another object

Other <object>

Create

Change

Display

Copy from...

- Backup action ---

Save

Hold

Generate

- Other actions (for example import/export functions) ---

Print

- Closing action ---

Delete

Exit

2. Data screens:

- Actions for choosing another object

Other <object>

Copy from...

- Backup action ---

Save (or Post)

- Other actions (for example import/export functions) ---

Print

- Closing action

Exit

Edit

Actions which edit the current object component - **obligatory**

- Select action
 - Select all
 - Deselect all
 - Select block
 - Choose
- Editing action ---
 - Cut
 - Copy
 - Paste
 - Move
- Other actions ---
 - Insert line
 - Delete line
 - Sort
 - Delete
- Reset action ---
 - Cancel

Other Menus

Other Menus

Goto

Object components of the current object - **obligatory**

Overview

Heder

Next <object component>

Previous <object component>

Other <object component>...

Back

Extras

Functions which supplement the current object/object component and are not permanently required - **optional**

View

Functions which display the current object/object component in different views - **only if required**

Settings

Functions which allow the user-specific defaults of parameters - **only if required**

Utilities

Functions which not only edit the object but are valid for all objects - **only if required**

Your Own Additional Menus

only if required

Environment

Tasks or reports of other application areas - **optional**

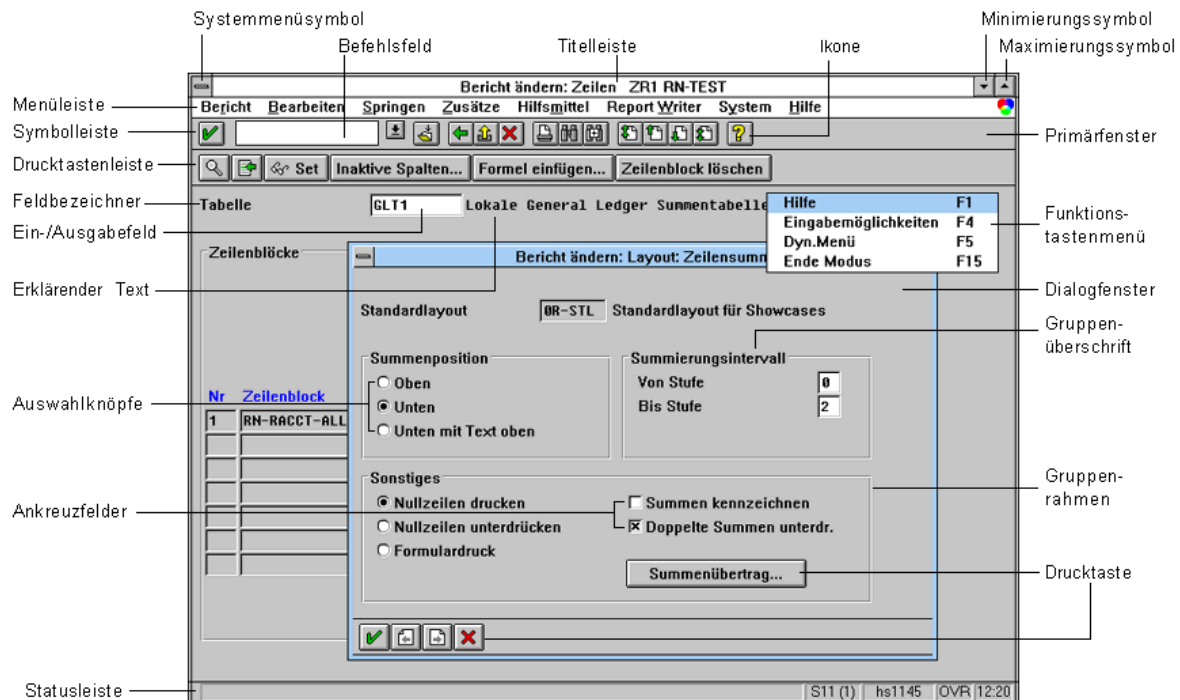
<Application>

Menu bar options at higher-level application level - **optional**

Graphical Overview of SAP GUI Terms (German/English)

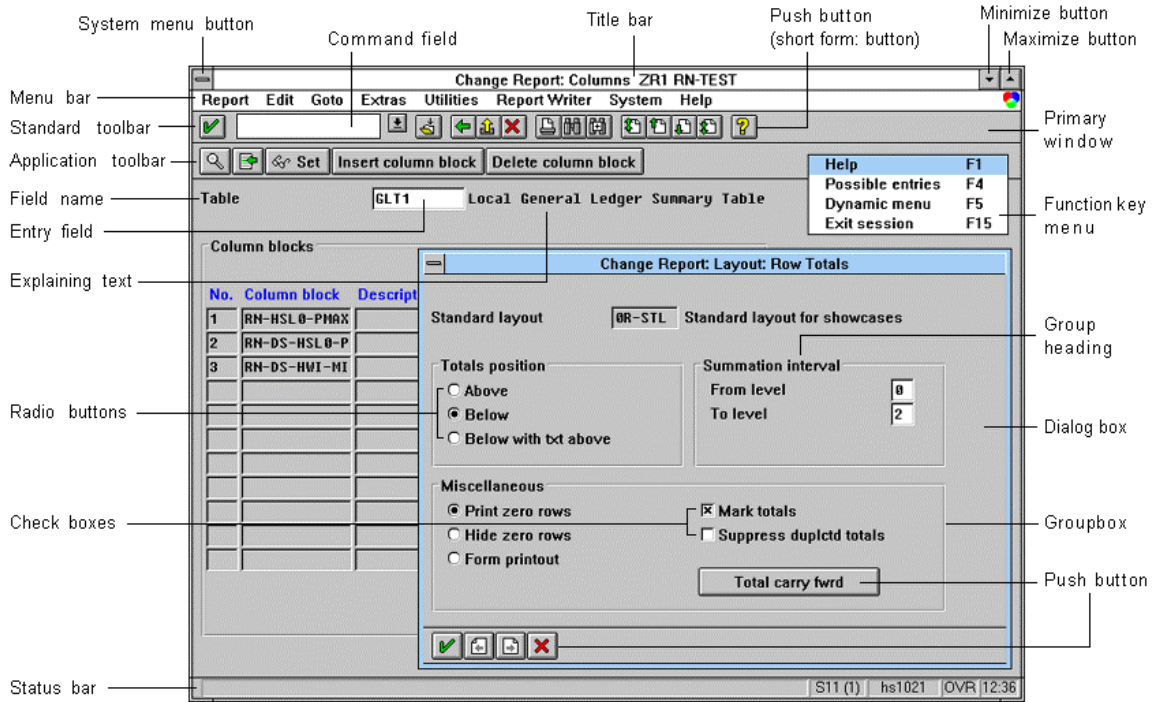
The following graphics show important SAP GUI terms in German and English.

German



Graphical Overview of SAP GUI Terms (German/English)

English



Glossary

[Abbreviations \[Seite 691\]](#)

[Access character \[Seite 701\]](#)

[Application level \[Seite 699\]](#)

[Application area menu \[Seite 700\]](#)

[Application toolbar \[Seite 714\]](#)

[Cascading menu \[Seite 733\]](#)

[Checkbox \[Seite 694\]](#)

[Clipboard \[Seite 761\]](#)

[Command field \[Seite 703\]](#)

[Command list \[Seite 704\]](#)

[Confirmation prompts \[Seite 752\]](#)

[CUA \[Seite 708\]](#)

[Dialog box \[Seite 709\]](#)

[ENTER pushbutton \[Seite 719\]](#)

[Error message \[Seite 721\]](#)

[EU 90/270/EWG \[Seite 720\]](#)

[Fast path \[Seite 710\]](#)

[Fields \[Seite 723\]](#)

[Field name \[Seite 722\]](#)

[Function \[Seite 725\]](#)

[Function key \[Seite 727\]](#)

[Function key menu \[Seite 728\]](#)

[Function name \[Seite 726\]](#)

[Group box \[Seite 729\]](#)

[Group heading \[Seite 730\]](#)

[Highlighting \[Seite 731\]](#)

[Hyphenation \[Seite 758\]](#)

[Initial screen \[Seite 718\]](#)

[Input/output fields \[Seite 716\]](#)

[ISO 9241-10 \(Dialogue principles\) \[Seite 732\]](#)

[Levels \[Seite 715\]](#)

[Macintosh \[Seite 734\]](#)

Glossary

[Main menu level \[Seite 748\]](#)

[Menu \[Seite 738\]](#)

[Menu bar \[Seite 739\]](#)

[Menu Painter \[Seite 740\]](#)

[Message \[Seite 736\]](#)

[Message type \[Seite 737\]](#)

[Modal dialog box \[Seite 741\]](#)

[Modeless dialog box \[Seite 693\]](#)

[Motif \[Seite 743\]](#)

[Multiple-selection list \[Seite 735\]](#)

[Navigation \[Seite 745\]](#)

[Possible entries pushbutton \[Seite 759\]](#)

[Primary window \[Seite 747\]](#)

[Pull-down menu \[Seite 692\]](#)

[Pushbutton \[Seite 711\]](#)

[Pushbutton with a fixed position \[Seite 712\]](#)

[Pushbutton with no fixed position \[Seite 713\]](#)

[Prompts \[Seite 690\]](#)

[Radio button \[Seite 702\]](#)

[Required-entry fields \[Seite 744\]](#)

[SAP matrix \[Seite 749\]](#)

[Screen \[Seite 705\]](#)

[Screen Painter \[Seite 750\]](#)

[Screen type \[Seite 706\]](#)

[Scroll bar \[Seite 707\]](#)

[Secondary window \[Seite 751\]](#)

[Session \[Seite 742\]](#)

[Single-selection list \[Seite 717\]](#)

[Standard toolbar \[Seite 754\]](#)

[Standards and guidelines \(national, international\) \[Seite 746\]](#)

[Status bar \[Seite 753\]](#)

[Table \[Seite 755\]](#)

[Task \[Seite 695\]](#)

[Task function \[Seite 697\]](#)

[Task level \[Seite 696\]](#)

[Title \[Seite 756\]](#)

[Title bar \[Seite 757\]](#)

[Window \[Seite 724\]](#)

[Windows \[Seite 760\]](#)

[Work area \[Seite 698\]](#)

Prompts

Prompts

The system displays prompts to the user if he/she carries out an action which endangers the security or consistency of data.

Abbreviations

It is often necessary to abbreviate one or more words because there is insufficient space available. The abbreviations should be made according to the abbreviation rules specified in Webster's Ninth Collegiate. If the space is still insufficient, you can create even shorter words according to the abbreviation rules mentioned under [Abbreviations \[Seite 593\]](#). It might be necessary to abbreviate words in the following places:

- [title bar \[Seite 757\]](#)
- [menus \[Seite 692\]](#)
- [function names \[Seite 726\]](#) for pull-down options, [pushbuttons \[Seite 711\]](#) and [function keys \[Seite 727\]](#) (the space is often insufficient here)
- [field name \[Seite 722\]](#)
- [messages \[Seite 736\]](#)

Pull-down menu

Pull-down menu

A menu of various options which is displayed by choosing a menu option either from a menu bar or another pull-down menu. Depending on the type of the menu option chosen, either a function is initiated or a subordinate pull-down menu ([cascading menu \[Seite 733\]](#)) is displayed. The latter is indicated by an arrow pointing to the right which is located to the right of the pull-down menu next to the corresponding menu option.

Modeless dialog box

A [dialog box \[Seite 709\]](#) is called "modeless", if the user can continue to work in other [windows \[Seite 724\]](#) without having to close this dialog box first. There are so far no modeless dialog boxes in the R/3 System.

Checkbox

Checkbox

Checkboxes are graphic control elements with which you can choose as many elements as desired from a list of [fields \[Seite 723\]](#). If the user is to choose exactly one element of the list, [radio buttons \[Seite 702\]](#) are used.

Task

With a task, you generally process an actual object of an object class, for example, you create a customer master record. A task consists of various task functions, such as *Save*, *Delete* or *Select*.

Task level

Task level

The task level is the third [level \[Seite 715\]](#) in the standard system after the [main menu level \[Seite 748\]](#) and the [application level \[Seite 699\]](#). Here, you process business tasks or tasks that are self-contained from the programming point of view ([task \[Seite 695\]](#)).

Task function

Task functions are functions with which the actual object processing takes place within the task and which are available after calling the task.

Work area**Work area**

The work area is the area of a [primary window \[Seite 747\]](#) or a [dialog box \[Seite 709\]](#) in which the user processes his current object. This area is between the [application toolbar \[Seite 714\]](#) and the [status bar \[Seite 753\]](#).

Application level

The application level is the second [level \[Seite 715\]](#) of the standard system which is displayed to the user after the [main menu level \[Seite 748\]](#). All [tasks \[Seite 695\]](#) of the application are listed here in [menus \[Seite 738\]](#). There are as many application menus as SAP software applications.

Application area menu

Application area menu

The [application level \[Seite 699\]](#) menu.

Access character

Access characters are the underlined letters in the [menu bar \[Seite 739\]](#) and the [pull-down menus \[Seite 692\]](#) which are used for starting a task or for initiating an action when choosing a menu option via the keyboard and when using the [fast path \[Seite 710\]](#).

Radio button

Radio button

Radio buttons are graphic control elements with which you can choose exactly one element from a list of fields. If several fields are to be chosen at the same time, you then should use [checkboxes \[Seite 694\]](#).

Command field

The command field can be found in the [standard toolbar \[Seite 754\]](#) to the right of the `ENTER` pushbutton. By entering [access characters \[Seite 701\]](#) or transaction codes in this field, the user can call a task without having to choose menu options. By using the [pushbutton \[Seite 711\]](#) to the right of the field, he/she can call up a list of the last 20 commands entered in the R/3 System (in all sessions). This is the [command list \[Seite 704\]](#).

Command list

Command list

The command list is the list of the last 20 commands entered into the command field of the R/3 System in the different sessions. The user can call up this list via a [pushbutton \[Seite 711\]](#) to the right of the [command field \[Seite 703\]](#) and choose a command from it.

Screen

Every [primary window \[Seite 747\]](#) of a [session \[Seite 742\]](#) can be referred to as a screen. This term should be used if the task consists mainly of controlling processing by altering the contents of the work area of various SAP windows.

Screen type

Screen type

[Bildern \[Seite 705\]](#)

When [screens \[Seite 757\]](#) are named in the title bar, they are assigned to a certain type, depending on the processing of the task. Depending on the aspect you want to particularly emphasize when processing data on a screen, you can distinguish, for example, between overview screens and detail screens or various data screens or supplementary data screens.

Scroll bar

The scroll bar can be operated with the mouse and offers various scrolling possibilities (absolute and relative scrolling forwards and backwards). It is displayed over the entire length of the object you want to scroll through either vertically to the right of the object or horizontally directly beneath the object.

CUA

CUA

CUA is an abbreviation for **C**ommon **U**ser **A**ccess. These are IBM guidelines to support the standardization of user interfaces.

Dialog box

A dialog box is a smaller [window \[Seite 724\]](#) which is called from and inserted into a [primary window \[Seite 747\]](#). It is subordinate to the primary window and is used to support the main action in the primary window.

Fast path

Fast path

Fast path means that the user can work with the [menu \[Seite 738\]](#) without choosing a line of the [menu bar \[Seite 739\]](#) and the other lines of the [pull-down menus \[Seite 692\]](#) sequentially. Instead of choosing, for example, *Program* in the menu bar and *Open* in the corresponding pull-down menu with the mouse (or the keyboard), the user can also enter *.po* into the [command field \[Seite 703\]](#). The period informs the system that the fast path is being used. The characters stand for the selections. Each character ([access character \[Seite 701\]](#)) represents one selection. This character is underlined in the menus.

Pushbutton

Pushbuttons or buttons are graphic control elements upon which you can execute the [function \[Seite 725\]](#) linked to this key with a single mouse click. In the SAP System, they can be started via the keyboard as well as with the mouse. To do this, position the cursor on the key in question and press the `ENTER` key or [ENTER pushbutton \[Seite 719\]](#). Pushbuttons can contain text or graphic symbols.

Pushbutton with a fixed position

Pushbutton with a fixed position

Fixed position pushbuttons are either displayed in the [standard toolbar \[Seite 754\]](#) (non-application-dependent functions) or in the [application toolbar \[Seite 714\]](#) (application-specific functions) and are defined in the [Menu Painter \[Seite 740\]](#). The system determines the position of the pushbutton automatically.

Pushbuttons of the application toolbar: in addition to the [function name \[Seite 726\]](#), the [function key \[Seite 727\]](#) which is allocated to this function is displayed on the pushbutton (for example *F3=Back*).

Pushbuttons of the standard toolbar: these pushbuttons contain graphic symbols and are stationary. Their meaning is explained in the [status bar \[Seite 753\]](#).

Pushbutton with no fixed position

Pushbuttons with no fixed position are located in the [work area \[Seite 698\]](#) and have no link to function keys. With a few exceptions (for example [Possible entries pushbutton \[Seite 759\]](#)), they contain text, and are defined in the [Screen Painter \[Seite 750\]](#).

Application toolbar

Application toolbar

The application toolbar is the area in the upper part of the window which covers the entire [window \[Seite 724\]](#) width below the [standard toolbar \[Seite 754\]](#). In this area, the [pushbuttons \[Seite 711\]](#) are displayed side by side in a pre-determined order.

Levels

The R/3 System is divided into three distinct levels. After the logon, you pass through them successively when first accessing a [task \[Seite 695\]](#). The level to which a [screen \[Seite 705\]](#) belongs is reflected in its [title \[Seite 756\]](#) and in the structure of the [menu \[Seite 738\]](#). The levels in the R/3 System are referred to by the following names:

- [main menu level \[Seite 748\]](#),
- [application level \[Seite 699\]](#) and
- [task level \[Seite 696\]](#)

Input/output fields

Input/output fields

Input/output fields are [fields \[Seite 723\]](#) in the [work area \[Seite 698\]](#) of the window, in which the system can display data and in which the user can enter data.

They are generally allocated to [field names \[Seite 722\]](#), unless, it is a tabular display.

Single-selection list

A single-selection list is a control element which displays a list of alternatives to the user, the number of which can be variable and from which he can choose one element.

If the user is to choose several elements of the list at the same time, you then should use a [multiple-selection list \[Seite 735\]](#).

Initial screen

Initial screen

The initial screen is the first [screen \[Seite 705\]](#) of a [task \[Seite 695\]](#). After the object class and the action have been chosen, the user can specify the object he/she wants to process on the initial screen.

ENTER pushbutton

In the [primary window \[Seite 747\]](#), you can see a [pushbutton \[Seite 711\]](#) with a tick on it on the left edge of the [standard toolbar \[Seite 754\]](#). This key has the [function \[Seite 725\]](#) which is generally used most for the entire [window \[Seite 724\]](#). Usually, this is the validation of the user input data or the display of the next [screen \[Seite 705\]](#).

EU 90/270/EWG

EC guidelines for designing workstations. In the appendix of these guidelines, the following criteria are named as minimum rules for the man-machine interface:

1. the software has to be appropriate to the activity to be performed,
2. it must be user-friendly and adaptable to the knowledge and experience of the user, if necessary,
3. the systems must provide information about the processing which is taking place,
4. they must display the information in a format and at a pace which is appropriate for the users,
5. the principles of ergonomics are to be applied in particular to the evaluation of information by human beings.

Error message

Error messages should help the user to recognize the cause of errors to be able to continue to work correctly. Depending on the [message type \[Seite 737\]](#) and whether they were issued from a [primary window \[Seite 747\]](#) or a [dialog box \[Seite 709\]](#), they are displayed in the [status bar \[Seite 753\]](#) or in a dialog box ([message \[Seite 736\]](#)).

Field name

Field name

A field name describes the possible contents of the following input or output field in a few words. It is placed in front of this field separated by a blank.

Field names are defined in the data elements in the Data Dictionary. They can be abbreviated according to the rules determined therein ([abbreviations \[Seite 691\]](#)).

Fields

See [Input/output fields \[Seite 716\]](#).

Window

Window

A window is an area on the screen which is surrounded by a border. A distinction is made between [primary window \[Seite 747\]](#) and [secondary windows \[Seite 751\]](#). This term is to be used every time you refer to the interaction element of the graphical user interface, regardless of the actual contents of the window.

Function

Functions can be started by selecting the entry corresponding to it in a pull-down menu, its [function key \[Seite 727\]](#) or its [pushbutton \[Seite 711\]](#). The corresponding processing is then carried out by the system.

Function name

Function name

Function names are used for the lines in the [pull-down menu \[Seite 692\]](#), on the [pushbuttons \[Seite 711\]](#) and in the window with the function key settings. For commonly used functions, standard function names have been defined.

Function key

[Functions \[Seite 725\]](#) can be allocated to the function keys available on the keyboard. Some function keys are by default always reserved for certain functions, others may not be allocated other functions if the allocated functions are inactive. The remaining function keys can be allocated any function.

Function keys are not the same as [pushbuttons \[Seite 711\]](#): pushbuttons are on the screen, function keys are on the keyboard.

Function key menu

Function key menu

The function key menu is displayed at the cursor position if you click the right mouse button. It shows a list of the available [function keys \[Seite 727\]](#) (including `ENTER`) as a pop-up menu. From this list, you can choose a [function \[Seite 725\]](#) using the mouse or the keyboard.

Group box

Group boxes enclose groups of [fields \[Seite 723\]](#) or other control elements to make them stand out against other groups or single elements. Group boxes are an important screen layout device. Group boxes consist of the border which encloses the control elements and a [group heading \[Seite 730\]](#) at the top of the group box.

Group heading

Group heading

The group heading is a part of the [group box \[Seite 729\]](#) - it is aligned at the upper left of the box - and names the group of control elements in the box. In some cases, there may be no group heading.

Highlighting

Highlightings are intended to improve the readability of data in the [work area \[Seite 698\]](#) of the [window \[Seite 724\]](#). For the R/3 System, a number of ergonomically useful ways of highlighting have been defined.

ISO 9241-10 (Dialogue principles)

ISO 9241-10 (Dialogue principles)

The first of 16 planned parts of the ISO guidelines for the layout of workstations to be completed (ISO = International Standards Organization). Part 10 deals with online principles and names seven criteria:

1. suitability for the task,
2. self-descriptiveness,
3. controllability,
4. conformity with user expectations,
5. error tolerance,
6. suitability for individualization,
7. suitability for learning.

Cascading menu

Cascading menus are subordinate [pull-down menus \[Seite 692\]](#) which are called from a pull-down menu. Cascading menus can be multi-level; however, you should not burden the user with more than three levels. In addition, such functionality should possibly be hidden in subordinate menus that are only rarely used or are only used by experts. You can see that cascading menus exist by the presence of an arrow to the right at the right edge of a pull-down option.

Macintosh

Macintosh

The Macintosh computer by Apple was the first PC (1984) with an entirely graphical user interface. Important basics for this were developed at XEROX; Apple has simplified and enhanced them for the mass market. Apple was also the first company to issue guidelines for the task layout (Human Interface Guidelines). These guidelines guaranteed a largely standardized task screen on the Macintosh computer.

Multiple-selection list

A multiple-selection list is a control element which offers the user a variable length list of alternatives from which he/she can choose as many elements as desired. You choose elements by pressing radio buttons.

If the user can only choose at the most one element from the list, you should use a [single-selection list \[Seite 717\]](#).

Message

Message

Messages are, if they are issued by a [primary window \[Seite 747\]](#), either written into its [status bar \[Seite 753\]](#) or in a [dialog box \[Seite 709\]](#) which is opened separately. If they are issued by a dialog box, an additional dialog box is always opened for its display ([error message \[Seite 721\]](#)).

Message type

There are five different types of [message \[Seite 736\]](#) in the R/3 System:

S (Success),

I (Information),

W (Warning),

E (Error) and

A (Abend) messages.

The message type affects the style of its output and the further processing.

Menu

Menu

[Aktionsmenüs \[Seite 692\]](#)

[Aktionsmenü \[Seite 692\]](#)

Menus are control elements which offer the user a number of options which, when chosen, initiate the execution of an action by the system. An action can be the opening of a subordinate [pull-down menu \[Seite 739\]](#).

There are two types of menus, the menu bar and the pull-down menu. The layout of the menu bar or the pull-down menus has been defined for each of the [levels \[Seite 715\]](#) of the R/3 System.

Menu options are selected either with a single mouse click or by positioning the cursor and by pressing the `ENTER` key.

Menu bar

The menu bar is located directly below the [title bar \[Seite 757\]](#) above the [work area \[Seite 698\]](#) of the window.

Choosing a menu item opens the allocated [pull-down menu \[Seite 692\]](#) below the item.

"Menu bar" also refers to a [function \[Seite 725\]](#) which activates the menu bar. After the activation, you can position the cursor on the required menu item using the arrow keys and the required function can be started by pressing `ENTER`.

Menu Painter

Menu Painter

The Menu Painter is a component of the ABAP/4 Development Workbench. In the Menu Painter, you can define the following interface elements with their associated attributes:

- [title \[Seite 756\]](#)
- [menu bar \[Seite 739\]](#) and ☐ [pull-down menus \[Seite 692\]](#)
- [function keys \[Seite 727\]](#) (predefined), [pushbuttons \[Seite 711\]](#) and ☐ [application toolbar \[Seite 714\]](#) / [standard toolbar \[Seite 754\]](#)

Modal dialog box

A modal dialog box is a [dialog box \[Seite 709\]](#) in which the user first has to acknowledge (**ENTER**) or cancel (**F12**) before he/she can continue to work in the subordinate [window \[Seite 724\]](#). In the R/3 System, all dialog boxes are modal.

Session

Session

[Anwendungsfunktion \[Seite 697\]](#)

[Statusleiste \[Seite 753\]](#)

A session is represented by a [window \[Seite 724\]](#) in which a certain task can be processed. When logging on to the R/3 System, the system opens the first session with a window. You can open up to nine sessions at the same time. The number of the current session is displayed in the status bar.

Motif

Motif or OSF/Motif is an Open Software Foundation (OSF) standard for graphical user interfaces which is independent of hardware and producers. It is primarily used by computers with the UNIX operating system.

Required-entry fields

Required-entry fields

Required-entry fields are [input/output fields \[Seite 716\]](#) which are indicated by "?". A [window \[Seite 724\]](#) can only be processed successfully, if all of its required-entry fields are filled in. In contrast, you do not necessarily have to make an entry in optional-entry fields.

Navigation

Navigating in the R/3 System is intended to give the user as much freedom as possible when moving between [screens \[Seite 705\]](#). A number of standard navigation functions are offered for this purpose (for example *Back*, *Exit* and *Cancel*). In addition to this free navigation, which is possible by choosing those [functions \[Seite 725\]](#) or by choosing absolute destinations, it can also be useful to guide the user through a sequence of screens. However, the user should always have the option to navigate freely.

Standards and guidelines (national, international)

Standards and guidelines (national, international)

In the field of software ergonomics, a number of national and international standards or guidelines have been established for the man-machine interface. They sometimes refer to the work station design only, but sometimes to the design of the user interface as well. The German national standard 66234 part 8 deals with the user interface (5 criteria). It is supplemented and superceded by the international [ISO standard 9241-10 \[Seite 732\]](#) (7 criteria). For Europe, there are the guidelines [EU 90/270/EWG \[Seite 720\]](#) which will soon also become the German national standard.

Primary window

Primary windows are the main windows of a [task \[Seite 695\]](#), that is, the [windows \[Seite 724\]](#) in which the main action of the task is carried out. Within a [session \[Seite 742\]](#), there is always exactly one primary window at the same time on the screen. Since a maximum of nine different sessions can be created in the R/3 System, a maximum of nine primary windows can be opened at the same time on the screen.

Primary windows can call [secondary windows \[Seite 751\]](#) or be replaced by another primary window.

Main menu level**Main menu level**

The main menu level is the first [level \[Seite 715\]](#) of the R/3 System directly after the logon. The user can choose his/her required application here.

SAP matrix

Here, an SAP matrix refers to a table structure ready for input containing [fields \[Seite 723\]](#), lines and columns as objects which can be selected. The user can select as many lines, columns and fields as desired.

Screen Painter

Screen Painter

[Drucktasten \[Seite 711\]](#)

The Screen Painter is a component of the ABAP/4 Development Workbench. In the Screen Painter, you can define the following interface elements with their associated attributes:

- [input/output fields \[Seite 716\]](#)
- [field names \[Seite 722\]](#)
- [checkboxes \[Seite 694\]](#)
- [radio buttons \[Seite 702\]](#)
- [Group boxes \[Seite 729\]](#)
- subscreens
- pushbuttons with no fixed position

Secondary window

Secondary windows are windows that are permanently linked to a [primary window \[Seite 747\]](#). They can only be created if they are sent by a primary window or by another secondary window. In secondary windows, peripheral processing is executed, such as processing supplementary data which is only needed in exceptional cases.

[Modal dialog boxes \[Seite 741\]](#) are the only kind of secondary windows in the R/3 System.

Confirmation prompts

Confirmation prompts

Confirmation prompts should always be displayed, if data could be lost. Various types of queries are distinguished: confirmation prompts, action confirmations and selection between two alternatives. Function modules have been implemented for the individual categories which guarantee that the queries are correctly displayed in a [dialog box \[Seite 709\]](#) and that the required standardized processing steps are carried out.

Status bar

[Meldungen \[Seite 736\]](#)

The status bar is a part of the [primary window \[Seite 747\]](#) and located at its lower edge. It contains an output field which covers almost the entire width of the window. In this field, the SAP System displays certain messages types ([message type \[Seite 737\]](#)) that are issued by a primary window. Other fields provide information on the status of the R/3 System. The SAP logo can be found to the very right.

Standard toolbar

Standard toolbar

The standard toolbar is located at the top of the window below the [menu bar \[Seite 739\]](#) and covers the entire width of the window. It only exists in [primary windows \[Seite 747\]](#). It contains a number [pushbuttons \[Seite 711\]](#) with a fixed position with symbols for non-application-dependent functions such as *OK/Continue*, *Save*, *Exit*, *Back*, *Cancel*, *Possible entries* and *Help*. In addition, it contains the [command field \[Seite 703\]](#) for entering commands into the R/3 System to the right of the [ENTER pushbutton \[Seite 719\]](#) which is on the far left.

Table

The term "table" refers to a certain way of displaying data on the screen. It also relates to certain functionality such as selecting and moving table entries.

Title

Title

The title of the [primary window \[Seite 747\]](#) and the [dialog boxes \[Seite 709\]](#) specify - as far as this can clearly be displayed - where the user is in the system. It is set up according to certain rules and written into the [title bar \[Seite 757\]](#).

Title bar

The title bar is located at the top of every R/3 System [Window \[Seite 724\]](#). The [title \[Seite 756\]](#) of the window is displayed in the center of the title bar.

Hyphenation

Hyphenation

For continuous text (for example in [messages \[Seite 736\]](#)), it can be useful to separate individual words to improve the structure of the text.

Possible entries pushbutton

The Possible entries pushbutton is displayed temporarily or permanently behind [input/output fields \[Seite 716\]](#) to which the system can list possible input values. The non-permanent possible entries pushbuttons are displayed as soon as the focus is set on an entry field which contains possible entries.

The Possible entries pushbutton is a [pushbutton \[Seite 711\]](#) in the [work area \[Seite 698\]](#) of the [primary window \[Seite 747\]](#) and the symbol is an arrow pointing downwards.

Windows

Windows

Windows is a Microsoft graphical user interface for computers using the MS-DOS operating system. The Windows NT version competes with UNIX and can be used on different platforms.

Clipboard

The clipboard is an area in which objects of every type can be buffered temporarily.