AB4 Editor

Fully-Functional Code Editor Control for Win32 Platforms

© Copyright 2005 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®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corp. in USA and/or other countries.

ORACLE® is a registered trademark of ORACLE Corporation.

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

LINUX is a registered trademark of Linus Torvalds and others.

Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.

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.

J2EE™ 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, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP, mySAP.com, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. MarketSet and Enterprise Buyer are jointly owned trademarks of SAP Markets and Commerce One. All other product and service names mentioned are the trademarks of their respective owners.

Disclaimer
Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Table of Contents

Table of Contents **3**

Brief Description of the AB4 Editor **4**

Main Features of the New Editor **5**

Features Matrix **8**

Description of Functionality **11**

1 Powerful, Fully-Customizable Highlighting for Predefined Keywords, Comments, and Strings 11
2 Language Schemes 11
3 Fully-Customizable Display Functionality 11
4 Different Types of Word Wrap 12
5 Different Cursor and Caret Styles 13
6 Active Status Bar 13
7 Edit Functionality 13
8 Drag & Drop Editing 14
9 Clipboard Ring 15
10 Block Selection of Text in Addition to Normal Selection 15
11 Multiple Clipboard Formats 15
12 Extended "Pretty Print" Functionality 16
13 Automatic Brace and Structures Matching, Current Scope Displaying 16
14 Powerful Find/Replace Functionality 17
15 Incremental Search (Forward/Backward) 17
16 Code Outlining/Structuring 17
17 Auto Completion 18
18 Code Hints 19
19 Quick Info 19
20 User-Defined Code Templates 20
21 Bookmarks 20
22 Scroll navigation 20
23 Printing with Previewing of documents in color 21
24 Window Splitter 21
25 Go to Line 22
26 Different Keyboard Schemes 22
27 External Files Handling 22
28 Export Functionality 22
29 XML Settings 23
30 Other 23

Brief Description of the AB4 Editor

The AB4 Editor is an ActiveX control that can be used to replace the standard source code editor (based on an SAP TextEditor) for any SAP system that supports ActiveX embedding. The reason for the replacement is that the standard RTF control, which the SAP TextEditor is based on, is not optimized for working with line-based text (as source code) and does not support modern code-editing functionality.

AB4 Editor Control is a line-based editor, specially developed for working with dynamically defined text blocks that ensure good performance during code highlighting. The editing functionality is optimized for programmers' needs, and is be fully adjustable because it is independent of standard components.

The AB4 Editor interface is optimized to work with the SAP frontend for Win32 compatible platform versions 4.6c and higher, which support SAP Enjoy Controls technology. The AB4 Editor is fully integrated into the SAP R/3 environment by a specially-designed ABAP wrapper and supports all the functionality available in the current editor. It is designed to replace the standard TextEditor in such transactions where the current code editor is used (ABAP Workbench - SE80, Function Builder - SE37, Report Builder - SE38, Class Builder - SE24, etc.) by replacing calls of the SAP TextEditor with calls of the AB4 Editor Control.

The AB4 Editor provides the following advantages for ABAP programmers:

- Enhanced speed of development
- Improved code quality
- Easier understanding of program source

All of those elements result in a reduction of resources spent by the company on application development.

Main Features of the New Editor

Use of the AB4 Editor provides the ABAP programmer with a fully functional environment, or Integrated Development Environment (IDE), for developing. It supports all the modern technologies of programming, such as functionality for powerful language syntax highlighting, outlining language structures, real-time code hints, and auto-completion of language structures and elements.

The AB4 Editor supports the easy adjustment of programming language syntax through XML configuration files, and the creation of new language schemes, which enables dynamic plugging of other languages (in addition to ABAP) without programming. By default, the editor has language schemes for support of ABAP, HTML, and C++ code.

The AB4 control includes a functional shell for creating and executing user-defined source templates, providing editor variables and interactive elements that speed up development, which reduces the amount of "routine coding."

The new editor supports extensive formatting functions and increases interactivity of programming in real time, greatly expanding the functions provided by the standard SAP ABAP editor (Pretty Print); for example: automatic code indentation, smart tabs, automatic brackets, automatic case and mistyping correction. All these actions are executed in real time, without the necessity of data transfer on backend.

AB4 Editor supports color printing with print preview, corresponding WYSIWING standards that facilitate better documentation creation, and the understanding of complex programs listings.

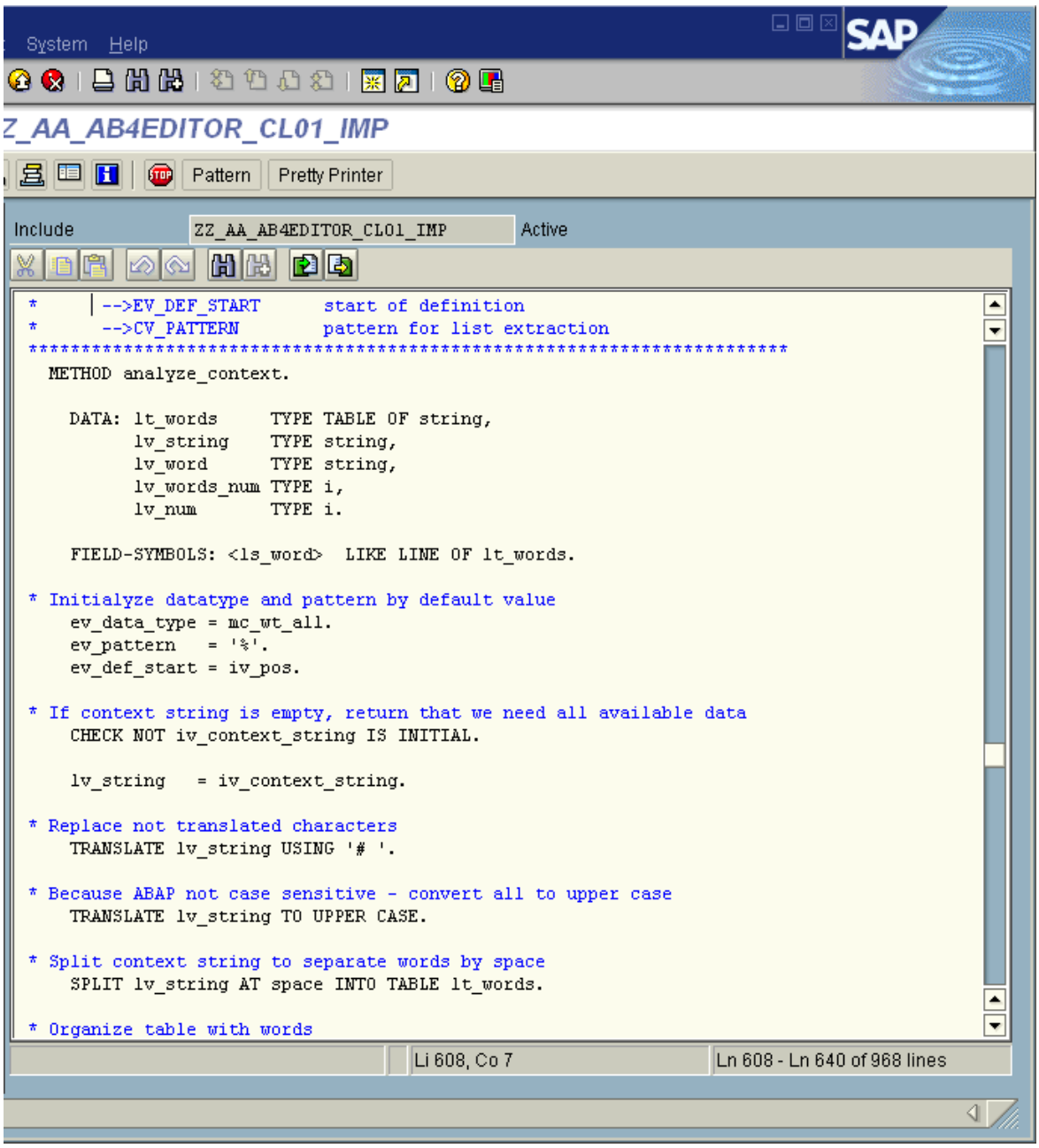

Figure 1: Current Editor for ABAP

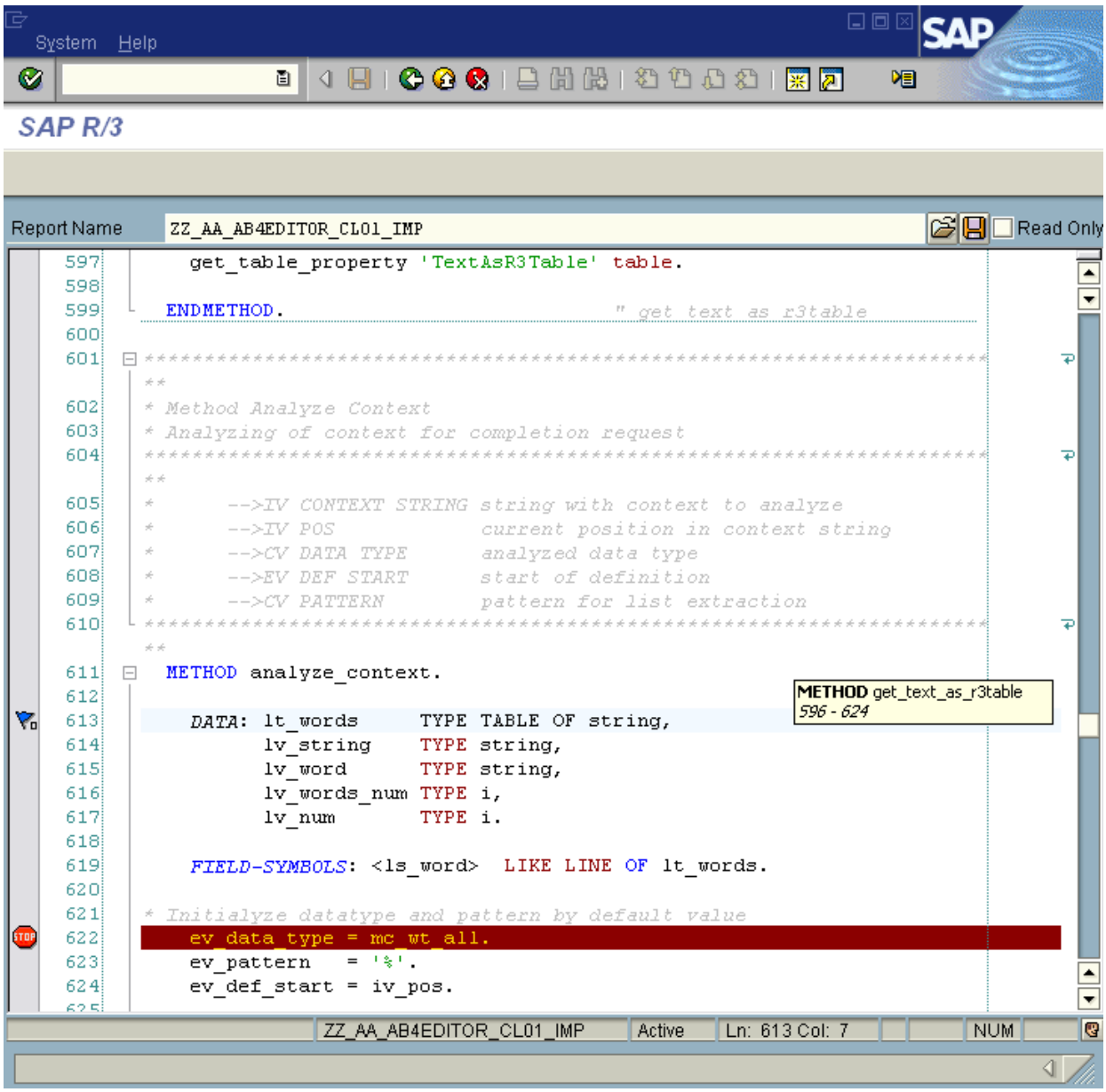

Figure 2: AB4 Editor Integrated in the SAP R/3 Environment

Features Matrix

Description of symbols used for feature indicating:

- ✘ - Does not exist
- ✓ - Exists
- ± - Partially exists

Feature Name	SAP ABAP Editor	AB4 Editor
Customizable syntax highlighting	±	✓
Support of programming languages	✘	✓
Display Functionality		
Changed lines indicator	✘	✓
Line numbers	✘	✓
Protected areas	✓	✓
Current line highlighting	✘	✓
Page width indicator	✘	✓
Structure separators	✘	✓
White space	✘	✓
Different types of word wrap	±	✓
Different cursor and caret styles	✘	✓
Active status bar	±	✓
Edit Functionality		
Comment/uncomment lines	✓	✓
Indent/un-indent lines	✓	✓
Case changing (upper/uower/capitalize/invert)	✘	✓
Sort lines	✘	✓
Join lines	✘	✓
Insert date/time in different formats	✘	✓
Copy append/cut append to clipboard	✘	✓
Drag and Drop editing	✓	✓
Clipboard ring	✘	✓
Plain text, RTF, and HTML clipboard format	±	✓
Block selection of text	✘	✓
Pretty Print		
Auto brackets	✘	✓
Auto indent	✘	✓
Smart tabs	✘	✓
Surround selection	✘	✓
Double rate on key repeat	✘	✓

Feature Name	SAP ABAP Editor	AB4 Editor
Runtime case correction	✗	✓
Customizable tabs	±	✓
Optimal fill	✗	✓
Caps lock correction	✗	✓
Auto correction of mistyping	✗	✓
Full text formatting	±	✓
Automatic brace and structures matching	✗	✓
Power Find/Replace Functionality		
Memory combo boxes with last typed items	✗	✓
Mark All function	✗	✓
Search with regular expressions	✗	✓
Incremental search	✗	✓
Code Outlining/Structuring		
Open/close code structures	✗	✓
Collapse comments	✗	✓
Format Tags outlining	✗	✓
Code Completion		
Auto completion of members and variables	✗	✓
Code hints	✗	✓
Quick Info and Hovering Tool tips	✗	✓
Statistic Analyze	✗	✓
Current Scope display	✗	✓
Code templates	±	✓
Bookmarks	✗	✓
Breakpoints	±	✓
Scroll Navigation	✗	✓
Print Functionality		
Color WYSIWYG printing	✗	✓
Print Preview	✗	✓
Adjustable footer/header	±	✓
Line numbers	✗	✓
Lines wrap	±	✓
Window Splitter	✗	✓
Go to line	±	✓
Different keyboard schemes	✗	✓
Export Functionality		
Export with syntax highlight to HTML	✗	✓
Export with syntax highlight to RTF	✗	✓
Export with syntax highlight to PDF	✗	✓
External Files Handling		

Feature Name	SAP ABAP Editor	AB4 Editor
Open/Save/Save As of external files	✓	✓
Save/Restore of breakpoints and bookmarks for files	✗	✓
Tracing external changes of opened files, and reloading if necessary	✗	✓
Automatic language determination	✗	✓
Automatic DOS/UNIX/MAC format determination	✗	✓
Unlimited file size	✓	✓
Auto Save document function	±	✓
Trim Trailing spaces on save	✓	✓
Leave Undo after Save	±	✓

Description of Functionality

1 Powerful, Fully-Customizable Highlighting for Predefined Keywords, Comments, and Strings

Adjustable highlighting colors provide the ability to adopt color schemes for any requirements. Keyword sets can also be displayed in bold, italic, or underline styles. Font faces and sizes are also completely customizable.

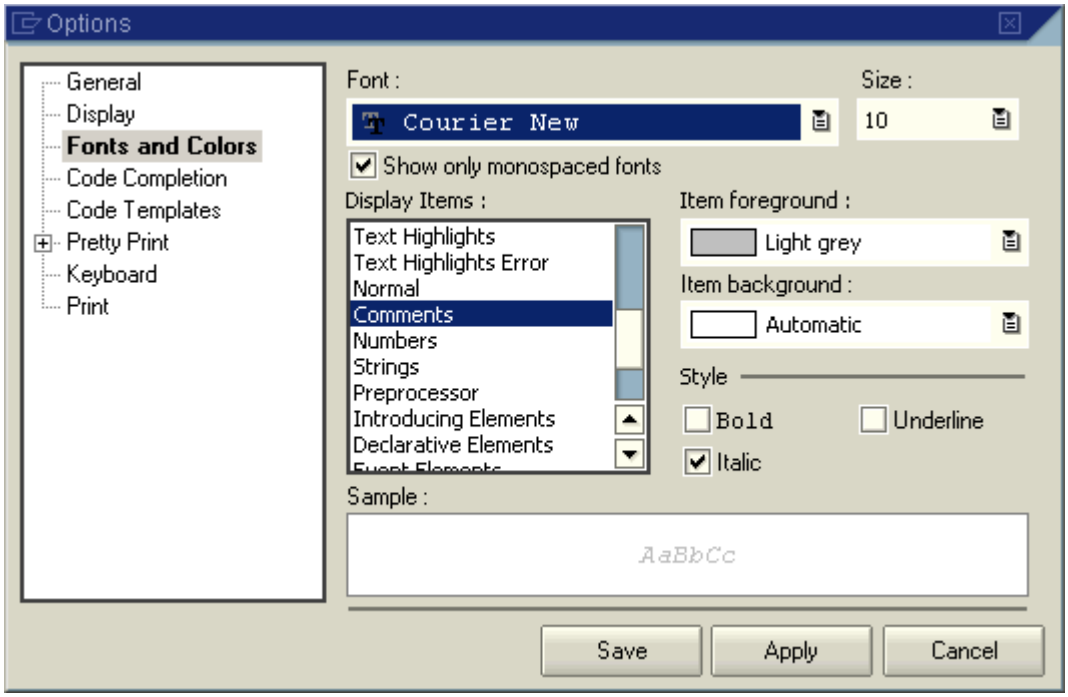

Figure 3: User settings for AB4 Editor. Customizing of color schemes

2 Language Schemes

Customizable programming language schemes allow the possibility of working with any programming language that has a maintained schema; support by default of ABAP, HTML, and C++ languages. This makes it possible to work in the same environment with ABAP, BSP, and Java, for example.

3 Fully-Customizable Display Functionality

The Editor supports various display settings:

a. The changed lines indicator shows you which parts of code you have changed during the last session and which changed parts were saved. Simplifies the changing of code known to be faulty or someone else's code.

b. Line numbers help to navigate though coding and is useful in word wrap mode to mark wrapped lines.
c. Protected areas used for defining part of the document as, for example, read-only; displays it using a read-only color and an additional symbol in the changed lines margin.
d. Current line highlighting makes it easy to find current cursor position in the document.
e. White space shows you hidden symbols such as spaces, tabs, and line breaks.
f. You can hide or display indicators, changed lines, numbers, and outlining margins depending on the amount of horizontal space needed.
g. Vertical and horizontal scrollbars can be hidden or displayed.
h. You can use page width to check the length of your lines and see if they are inside of a defined range.
i. Structure separators show you where high-level structures finish (function, subroutine, class, method, etc), separating them from one another and facilitating navigation.

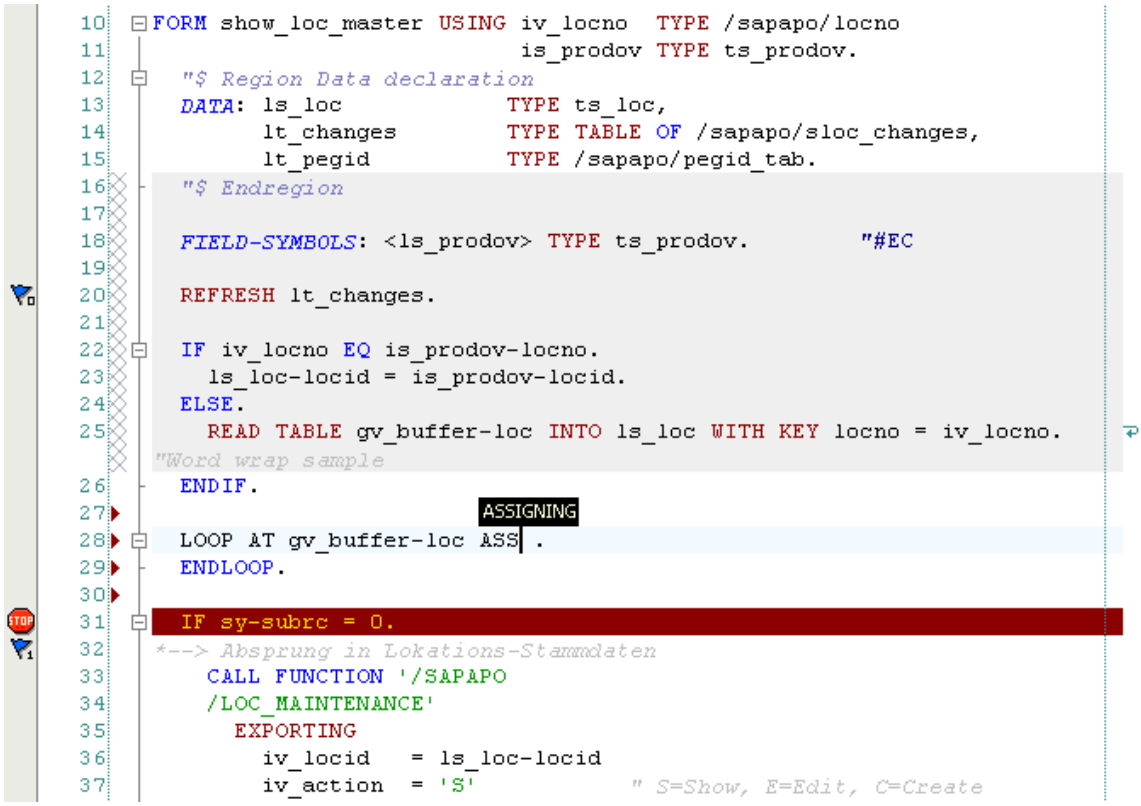

Figure 4: Visualization Possibilities in AB4 Editor

4 Different Types of Word Wrap

Customizable word wrap helps working with lines that are longer than the screen width or defined page width. There are several word wrap modes available:

- By column number

- By window edge
- By column number inserting hard break

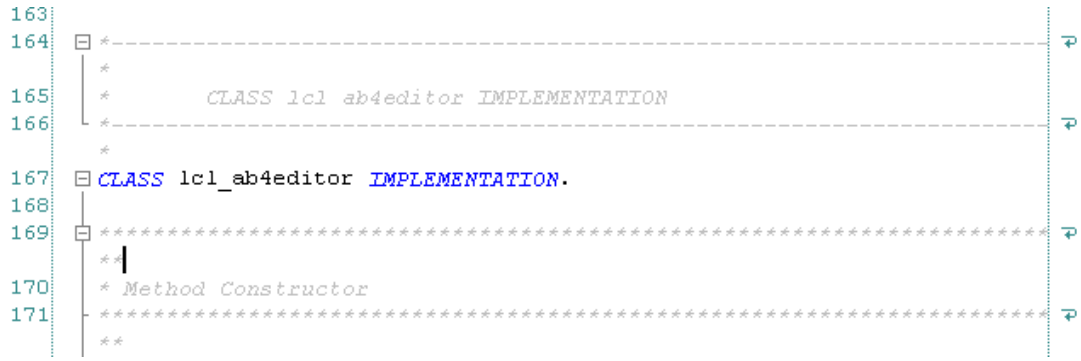

Figure 5: Word Wrap by Column Number Enabled in AB4 Editor

5 Different Cursor and Caret Styles

The Editor provides different cursor and caret styles for overwrite, insert modes, interactive search, and block selection that makes it easy to indicate different modes of operation/execution.

6 Active Status Bar

Active status bar: Panels indicate status of Caps Lock, Num Lock, and Overwrite modes. These modes can be changed by double-clicking on them. Double-clicking on the position panel will call the "Go to Line" dialog. Double-clicking on the panel with document name will copy it to clipboard.

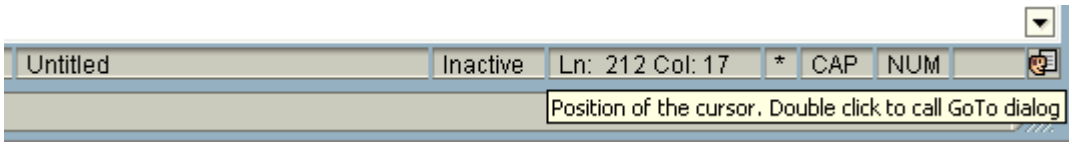

Figure 6: Editor Status Bar with Indicators, Active Panels, and User Settings Button

7 Edit Functionality

Edit functionality supports functions including:

a. Comment/uncomment lines
b. Indent/un-indent lines
c. Case-changing (upper/lower/capitalize/invert)
d. Sort lines
e. Join lines
f. Format selection by schema (as in SAP Pretty Print)
g. Insert date/time in different formats
h. Copy append/cut append to clipboard.

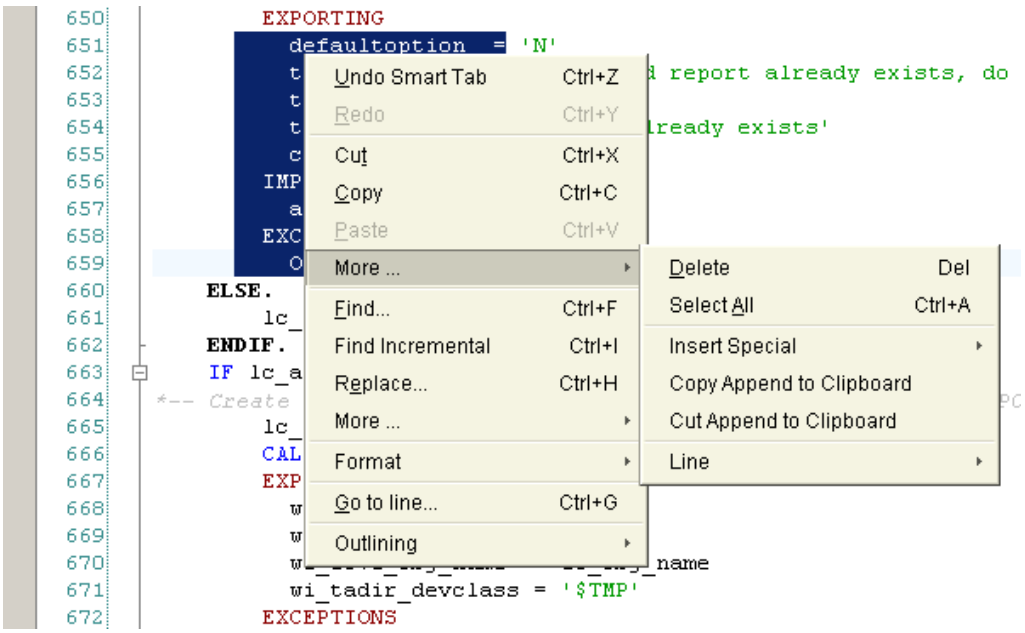

Figure 7: Context Menu for Edit of Selection

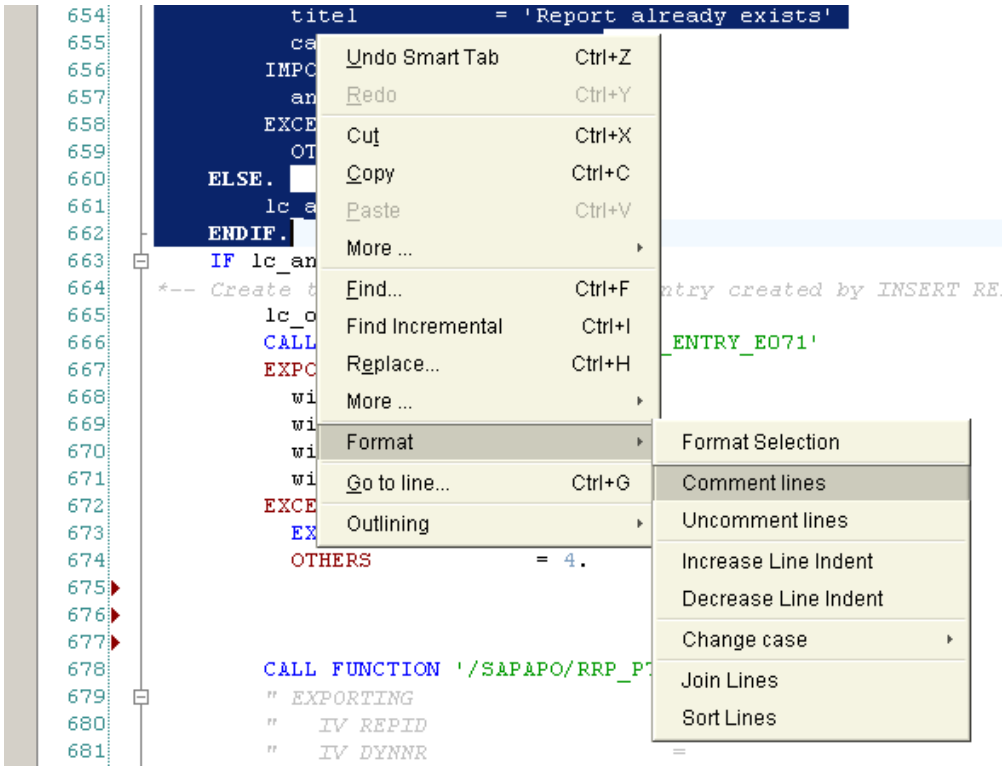

Figure 8 : Context Menu for Text Formatting

8 Drag & Drop Editing

The Editor supports Drag & Drop editing, providing another way for intuitive copying and pasting.

9 Clipboard Ring

When text is cut or copied, the fragments are stored in an editor clipboard known as the "Clipboard Ring," where you can pick and choose among them and paste back into your code. The clipboard contains up to 12 items, presented in historical sequence and moved with LIFO logic. This is a powerful feature for quickly copying several pieces of text to a new location or for recalling text that you inadvertently cut. The clipboard ring can remember block and normal items.

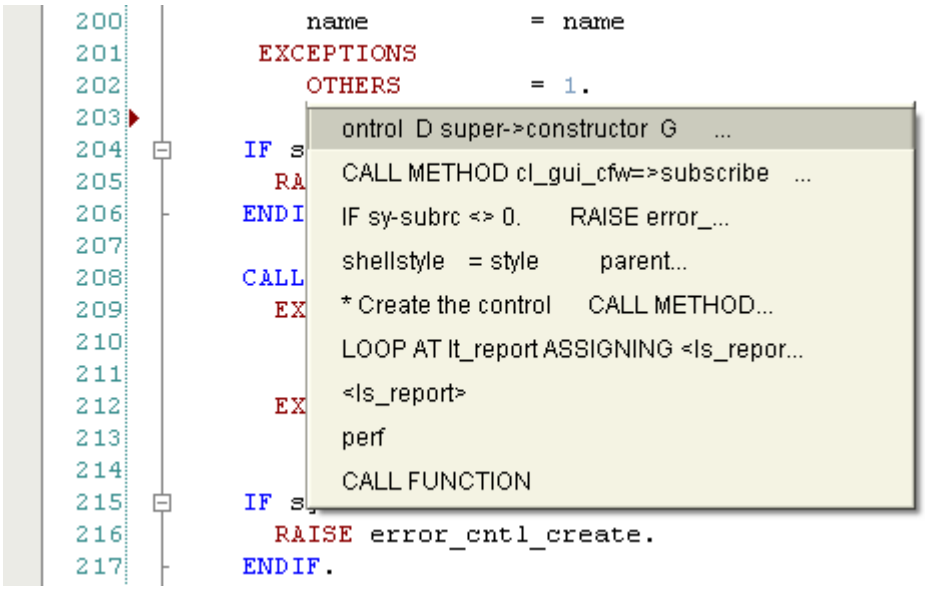

Figure 9: Extended Paste Menu Shows List of Text Items in Clipboard Ring

10 Block Selection of Text in Addition to Normal Selection

The block selection is supported by most of the common actions, which are executed with normal selection (copy, paste, indent, etc.).

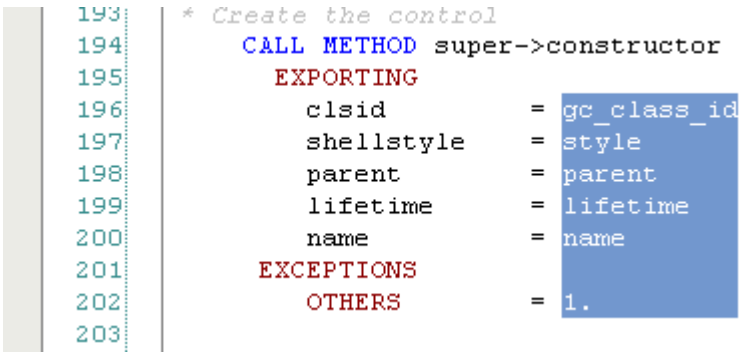

Figure 10: Block Selection of Text

11 Multiple Clipboard Formats

Support of multiple clipboard formats, such as Text, RTF, and HTML provides the possibility of WYSWING pasting text in Microsoft Word or Microsoft Outlook, for example. The Unicode text format is supported, allowing the copying and pasting of Unicode characters.

12 Extended "Pretty Print" Functionality

List of available runtime functions:

a. "Auto Brackets" automatically adds closed bracket on typed open bracket.
b. "Auto Indent" function recognizes language structures and applies runtime formatting of the code.
c. "Smart Tabs" optimize tab function in beginning of the lines.
d. "Surround Selection" helps insert surrounding brackets for selections.
e. "Double rate for key repeat" function speeds up navigation command execution on repeating.
f. "Format after every Paste" formats inserted text according to current indentation, avoiding manually doing so.
g. Runtime Case Correction for predefined keywords (Uppercase, Lowercase, Capitalize, and Like In Dictionary)
h. Customizable tab/space settings provide the option of choosing tab size and selecting that which you prefer.
i. Optimal fill watches for optimal use of tabs and spaces in space offsets.
j. Caps Lock Correction determines accidental usage of CAPS LOCK key and makes corrections.
k. "Auto Correction" function helps determine errors due to mistyping and corrects them (as in MS Word).

13 Automatic Brace and Structures Matching, Current Scope Displaying

Automatic brace matching gives you immediate feedback on misplaced brackets or open-ended code segments. Automatic brace matching recognizes all brackets types, quotes, and structures defined in language specification. When you type a closing bracket, both it and its matching bracket remain highlighted until you type another key or move the cursor (same works for open/close structures). This makes visually identifying code blocks much easier. In addition, this may help identify mismatched or missing brackets and structures. Double-clicking on open/close structure tag will navigate you to the matching close/open tag in the document.

The Current Scope feature enables you to see current parentheses and hierarchy scope when you navigate through the code, highlighting active bracket pairs and structures. The current scope feature also enables error highlighting (in red color) of mismatched brackets.

```
586     LOOP AT ftab_program_file.
587       lc_program_file = ftab_program_file.
588       CASE lc_program_file(9.
589       WHEN mc_trdir_identifier.
590         lc_datatype = mc_trdir_short.
591       WHEN mc_report_identifier.
592         lc_datatype = mc_report_short.
593       WHEN mc_text_identifier.
594         lc_datatype = mc_text_short.
595       WHEN mc_doc_identifier.
596         lc_datatype = mc_doc_short.
597       WHEN mc_thead_identifier.
598         lc_datatype = mc_thead_short.
599       WHEN OTHERS. " Actual contents of repor
600         CASE lc_datatype.
601         WHEN mc_trdir_short.
```

Figure 11: Runtime Highlighting of Matched Structures when Typing

14 Powerful Find/Replace Functionality

In addition to common features used in find/replace dialog extended functionality added:

a. Memory combo boxes with last typed items;
b. Mark All function. Marking all occurrences of search text in document with bookmarks;
c. Search with regular expressions.

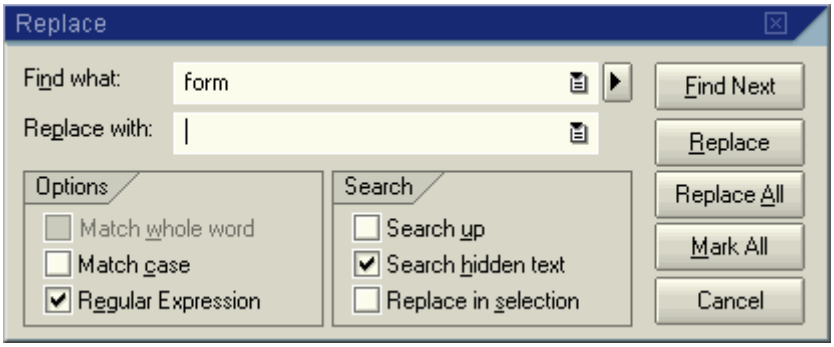

Figure 12: Extended Find/Replace Dialog

15 Incremental Search (Forward/Backward)

Runtime forward/backward search through document – when typing text, editor searches and selects for first occurrence of the search string on a letter-by-letter basis as you type characters.

16 Code Outlining/Structuring

The document text is analyzed at runtime using data predefined in language specifications and updated with each document change. Code outlining gives you a better document overview and the possibility of finding errors, such as unclosed structures, without checking or compiling. This is similar to the compression view of code in the old ABAP editor, where you could see collapsed if-endif blocks of code, to quickly determine of nested constructions were properly defined.

a. View code in hierarchy;
b. Collapse/expand of blocks;
c. Collapse to definition;
d. Multi-line comments outlining;
e. Newly-defined format tags for comments found in the middle of statements help create a structure in any part of the code.

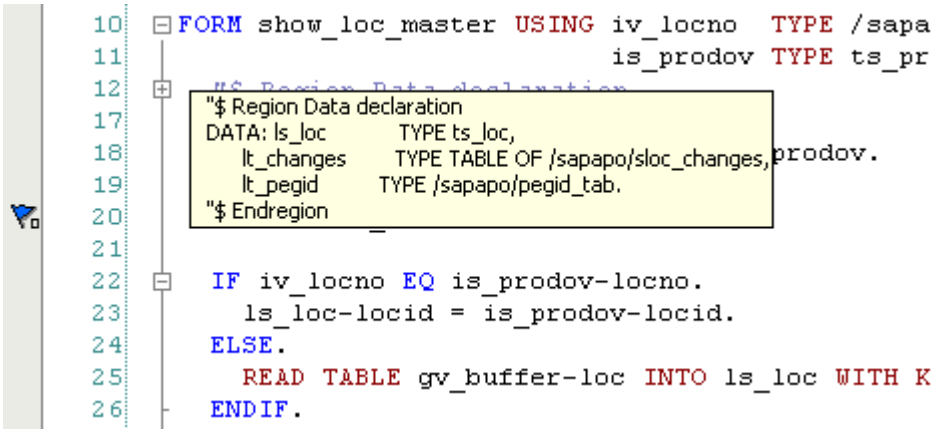

Figure 13: Code Outlining and Tool tip for Collapsed Area when Hovering

17 Auto Completion

You can use this feature to get a list of available functions, subroutines, macros, etc. by pressing Ctrl + Space in your code; after selecting one of the items it will be inserted. If there is only one item available, it will be inserted automatically without displaying the list. You can select an area where to search available items:

a. Internal dictionary with keywords.
b. External dictionary (load keywords list on request with support of icons and insert pattern requests).
c. Simple completion dictionary – suggested items extracted from already typed or loaded text.

Auto completion results can be also filtered in a list (when more than one type of data is displayed) by data type with additional filter buttons.

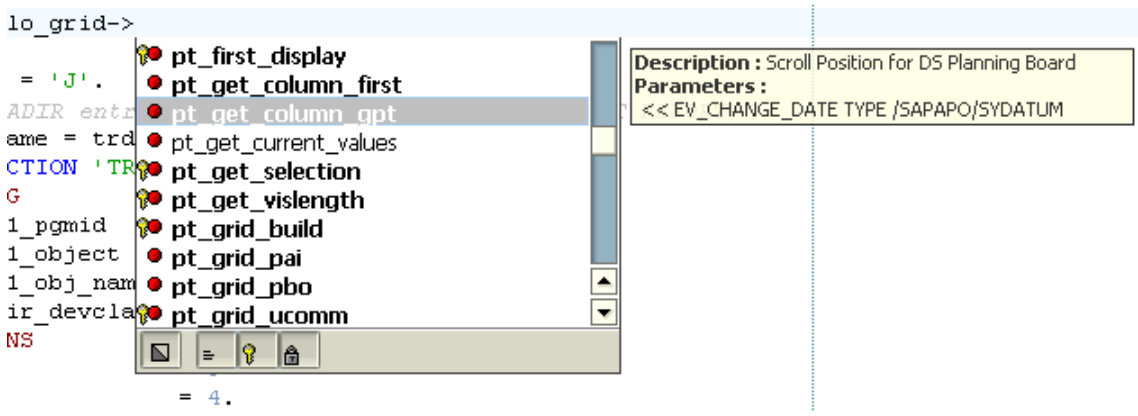

Figure 14: Example of Auto Completion

Completion of instance methods for class /SAPAPO/RRP_PT_ALV_GRID with method pattern insertion. Non-bolded style indicates inherited methods. Tool tip displays method parameters.

18 Code Hints

Code hints will help in runtime during typing, suggesting ways to examine word statistics, dictionaries, and templates. By pressing Tab or Enter, when Code Hint is shown, you can insert it into your code without typing the rest of the characters. The hint color shows how relevant a suggestion is.

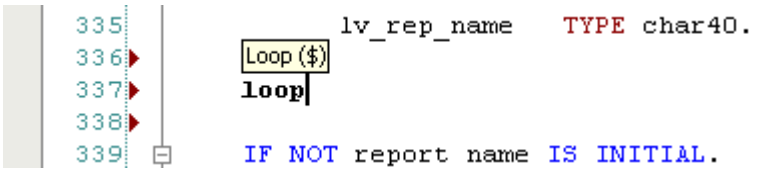

Figure 15: Code Hints: Editor Suggests Suitable Template

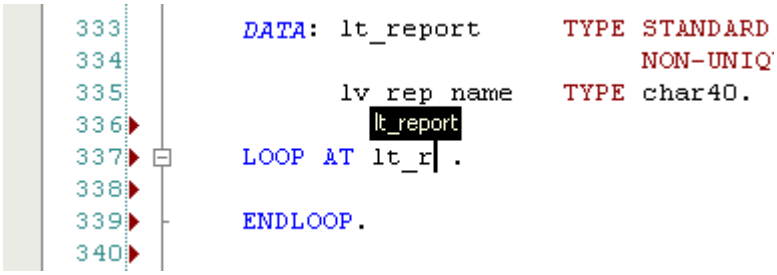

Figure 16: Code Hints: Editor Suggest Keyword; Inverse Color Indicates More than One Suggestion Available

19 Quick Info

Quick Info functions for any part of code. The external application receives a request for an information string and provides text for tool tip display. The tool tip recognizes bold and italic tags and can display multi-lines information.

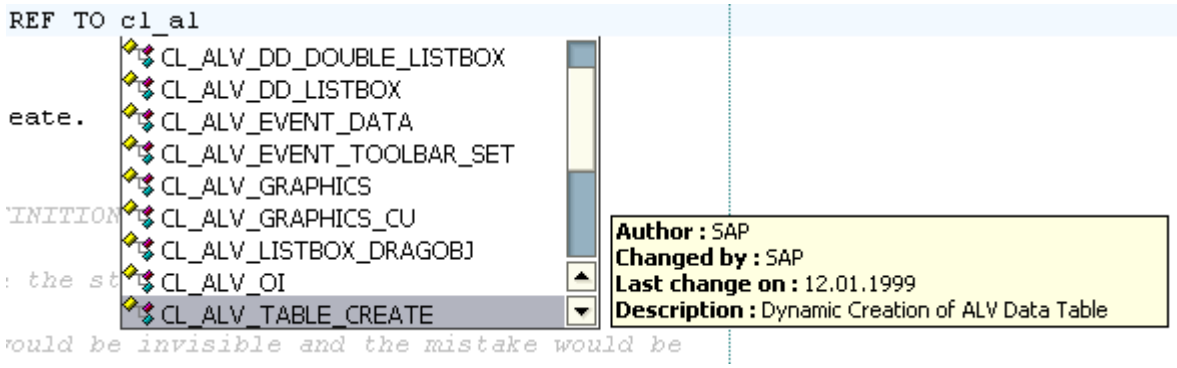

Figure 17: Quick Info for Class Completion

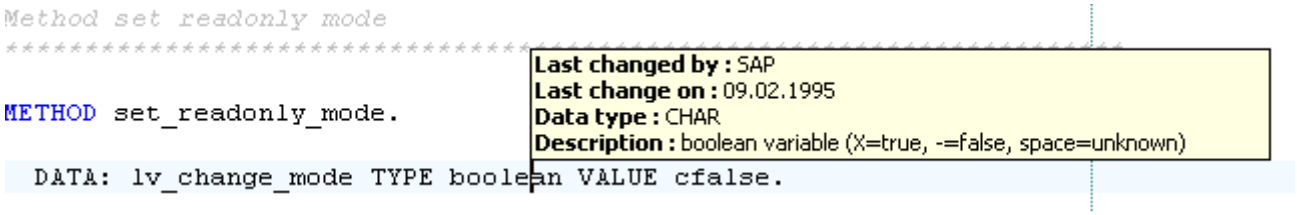

Figure 18: Quick Info for Data Type Called by Shift-Ctrl-Space in Text

20 User-Defined Code Templates

Powerful, user-defined templates with "Auto Expand" functionality greatly simplify typing of standard constructions. Supports such predefined tags as current date/time, clipboard content, and interactive selection.

21 Bookmarks

Support of named and unnamed bookmarks (as in MSVS) helps in navigation and code editing.

22 Scroll navigation

Scroll navigation provides you information about current position in code in a scroll tip, displaying current function, subroutine, class, method, etc. It also displays range of lines, which you currently scroll.

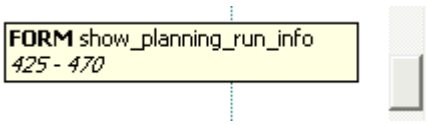

Figure 19: Scroll Tip Shows Current Position in Document

23 Printing with Previewing of documents in color

WYSIWING printing style provides more readable hard copies of documents. Printing supports user-defined footer and header text with predefined tags and different styles.

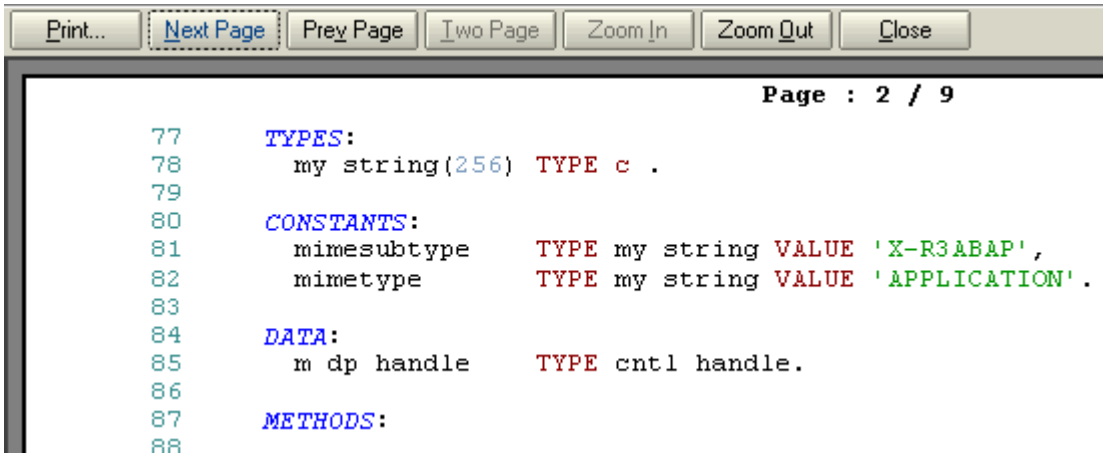

Figure 20: WYSIWYG Print Preview for Document Inside SAP R/3 Envirnment

24 Window Splitter

Using of window splitter helps you in work with different parts of the document simultaneously.

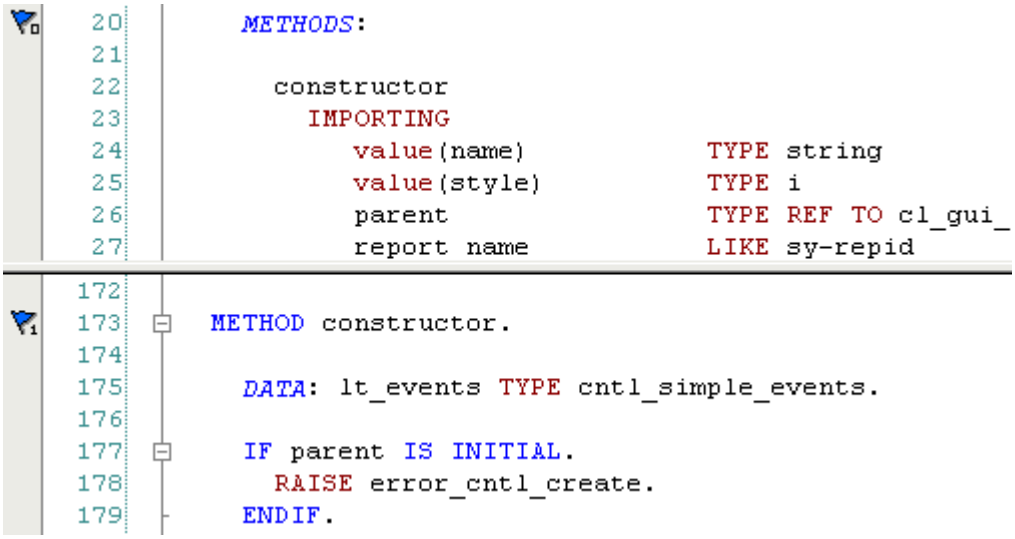

Figure 21: Using Horizontal Splitting

25 Go to Line

Editor includes go-to line functionality with absolute and relative navigation.

26 Different Keyboard Schemes

User-defined keyboard schemes provide the ability to define different keyboard schemes and add preferred and multiple shortcuts for functions.

27 External Files Handling

External files handling includes:

a. Automatic language determination
b. Automatic DOS/UNIX/MAC format determination
c. Tracing external changes of opened document, and reloading if necessary
d. Saving/restoring document info (breakpoints and bookmark)
e. Unlimited file size

28 Export Functionality

The editor supports WYSWING export functionality for most common text formats:

a. Export to HTML

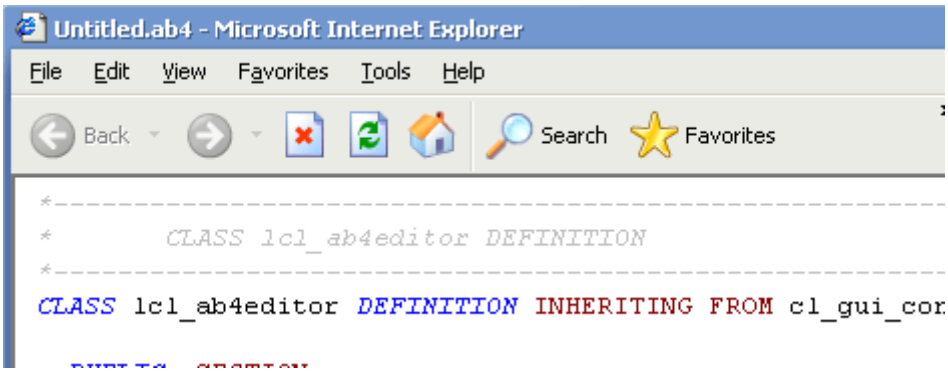

Figure 22: Internet Explorer Displays Document Exported to HTML

b. Export to RTF

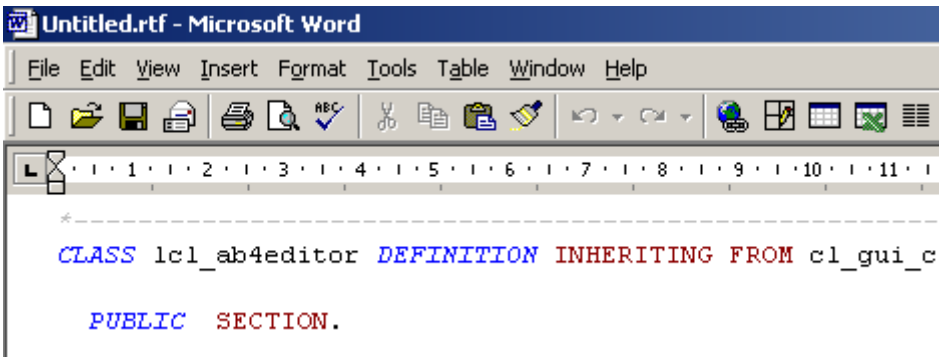

Figure 23: Microsoft Word Displays Document Exported to Rich Text Format (RTF)

c. Export to PDF

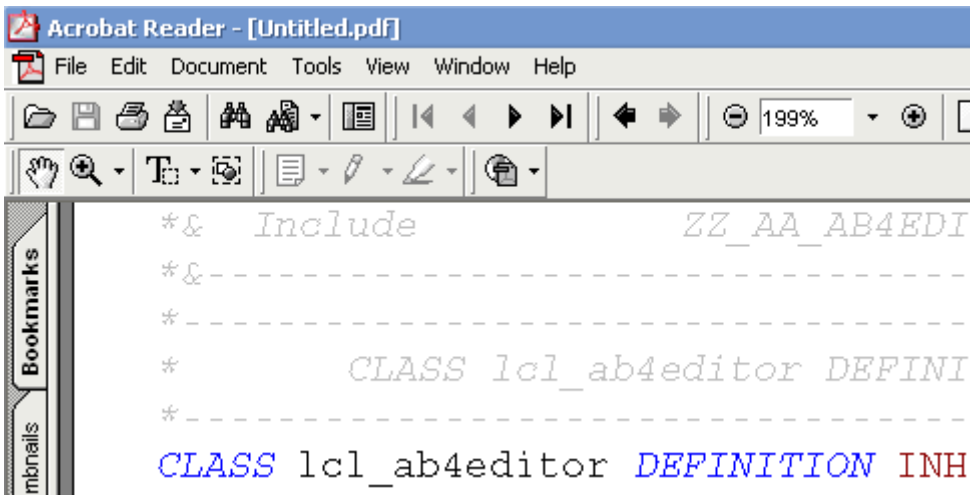

Figure 24: Acrobat Reader Displays Document Exported to PDF Format

29 XML Settings

All settings stored in XML format, easing export and import to other systems.

30 Other

Other functions provided by editor:

a. Auto Save functionality for modified external documents and files
b. Trim trailing spaces in lines on save
c. Leave undo after save provides ability to undo changes even after saving