

BlackBerry Software Development Kit

Version 2.5

AutoText API Reference Guide

BlackBerry Software Development Kit Version 2.5 AutoText API Reference Guide
Last modified: 6 May 2002

Part number: PDF-04634-001

At the time of publication, this documentation complies with RIM Wireless Handheld version 2.5.

© 2002 Research In Motion Limited. All Rights Reserved. The BlackBerry and RIM families of related marks, images and symbols are the exclusive properties of Research In Motion Limited. RIM, Research In Motion, 'Always On, Always Connected', the "envelope in motion" symbol and the BlackBerry logo are registered with the U.S. Patent and Trademark Office and may be pending or registered in other countries. All other brands, product names, company names, trademarks and service marks are the properties of their respective owners.

The handheld and/or associated software are protected by copyright, international treaties and various patents, including one or more of the following U.S. patents: 6,278,442; 6,271,605; 6,219,694; 6,075,470; 6,073,318; D445,428; D433,460; D416,256. Other patents are registered or pending in various countries around the world. Visit www.rim.net/patents.shtml for a current listing of applicable patents.

While every effort has been made to ensure technical accuracy, information in this document is subject to change without notice and does not represent a commitment on the part of Research In Motion Limited, or any of its subsidiaries, affiliates, agents, licensors, or resellers.

Research In Motion Limited
295 Phillip Street
Waterloo, ON N2L 3W8
Canada

Published in Canada

Contents

	About this guide.....	3
	Other documentation	3
CHAPTER 1	AutoText API Reference.....	5
	Index of functions	9
	Index	11

About this guide

The AutoText engine on the RIM Wireless Handheld is a powerful tool that enables users to type common phrases quickly. By entering pairs of strings into the AutoText list, users can associate replacement strings with original strings. Each time a user presses the SPACE key, the AutoText engine compares the last word that was typed to the list of original strings. If the word is found, it is replaced with the corresponding replacement string.

The AutoText application programming interface (API) enables you, as an application developer, to access the features of the AutoText engine from within your own application. By using this API, you enable users of your program to use the same keyboard shortcuts that they are accustomed to using in other applications that are designed for the handheld.

This guide includes overview information on the AutoText API, as well as a complete listing of all its functions.

Related documentation

Before using this guide, you should be familiar with the following documentation. These other resources can help you develop C++ applications for the RIM Wireless Handheld.

- *BlackBerry SDK Developer Guide*

The *BlackBerry SDK Developer Guide* explains how to use the BlackBerry SDK, with tutorials and sample code to demonstrate how to write handheld applications. For additional information, visit the BlackBerry Developer Zone at <http://www.blackberry.net/developers>.

About this guide

- *UI Engine API Reference Guide*

This guide includes a complete listing of all structures and functions used by the UI Engine.

- `README.txt`

The `README.txt` file is installed with the BlackBerry Software Development Kit. It provides information on any known issues and workarounds, as well as last-minute documentation updates and release notes.

Chapter 2 AutoText API Reference

The AutoText API enables you to perform the following tasks:

- add original string/replacement string pairs to the list of AutoText entries
- display the AutoText list screen from within your application
- create an Edit field that acts in the same way as any other Edit field in the system, except that it performs AutoText substitutions automatically.

To include AutoText API functions in your application, you must include `<AutoText.h>` in your code.

Functions

The following functions are listed alphabetically.

AutoText::AddAutoText	5
AutoText::AutoTextEdit	6
AutoText::AutoTextEntry	6

AutoText::AddAutoText

Adds an AutoText entry to the list.

```
HandleType AddAutoText(  
    char const * const pOriginalText,  
    char const * const pReplacementText)
```

Chapter 2: AutoText API Reference

Parameters	<code>pOriginalText</code>	Specifies the original text string for the AutoText entry.
	<code>pReplacementText</code>	Specifies the replacement text string for the AutoText entry.

Description `AddAutoText` adds an AutoText entry to the AutoText list. For automatic substitution to take place properly, the original text string should not contain any whitespace characters, or any of the characters in the following set: { , . ? ! : ; () }.

AutoText::AutoTextEdit

Constructor for UI Engine edit field.

Form 1: `AutoTextEdit()`

Form 2: `AutoTextEdit(
 char const * const pnewLabel,
 char * const pnewBuffer,
 int const newLengthofBuffer,
 int const newCharacterOffset = 0,
 int const Justify = LCD_LEFT_JUSTIFIED)`

Parameters	<code>pnewLabel</code>	Label associated with the field.
	<code>pnewBuffer</code>	A pointer to the application-defined buffer.
	<code>newLengthofBuffer</code>	Length of the buffer pointed to by <code>pnewBuffer</code> .
	<code>newCharacterOffset</code>	Starting cursor offset.
	<code>Justify</code>	Justification of the text in the buffer with respect to the screen.

Description The `AutoTextEdit` constructors automatically perform AutoText substitutions. The `AutoTextEdit` field can only be constructed with a buffer that is allocated by the application. To use a buffer allocated by the UI Engine, call Form 1 of the constructor, and call `SetBuffer` at a later time. The `AutoTextEdit` field is derived from the `Edit` field in the UI Engine. Refer to the *UI Engine API Reference Guide* for other functions that can be called.

AutoText::AutoTextEntry

Displays a list of AutoText entries in the system.

`void AutoTextEntry()`

Description Calling `AutoTextEntry` displays a screen that contains a list of the `AutoText` entries that are in the system. From this screen, the user can edit the phrases in the `AutoText` list manually. The member returns when the user chooses the **Close** menu item from the menu that is associated with the screen.

Index of functions

A

AutoTextEdit

AddAutoText(), 5

AutoTextEdit(), 6

AutoTextEntry(), 6

Index

A

adding an AutoText entry, 5

API functions, 5

AutoText

 overview, 3

AutoText entry

 adding, 5

 viewing, 6

V

viewing AutoText entries, 6



© 2002 Research In Motion Limited
Published in Canada