

SALT Release Notes

SALT Version 2.6

Dyalog Limited

Minchens Court, Minchens Lane
Bramley, Hampshire
RG26 5BH
United Kingdom

tel: +44(0)1256 830030
fax: +44 (0)1256 830031
email: support@dyalog.com
<http://www.dyalog.com>

Dyalog is a trademark of Dyalog Limited
Copyright © 1982-2016



*Dyalog is a trademark of Dyalog Limited
Copyright © 1982 - 2016 by Dyalog Limited.
All rights reserved.*

Version 2.6

Revision: 20160311_260

No part of this publication may be reproduced in any form by any means without the prior written permission of Dyalog Limited, Minchens Court, Minchens Lane, Bramley, Hampshire, RG26 5BH, United Kingdom.

Dyalog Limited makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Dyalog Limited reserves the right to revise this publication without notification.

SQAPL is copyright of Insight Systems ApS.

UNIX is a registered trademark of The Open Group.

Windows, Windows Vista, Visual Basic and Excel are trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

All other trademarks and copyrights are acknowledged.

Contents

1	ABOUT THIS DOCUMENT.....	1
1.1	Compatibility with Dyalog Versions	1
2	FUNCTIONAL CHANGES.....	2
2.1	Updates to Values	2
2.2	Modifier Values that Include Space Characters	2
2.2.1	Multiple Modifier Values	2
2.3	Performance of the Snap Function's Load File.....	2
2.4	Object Name Changes.....	3
2.5	New Function: Clean	3
2.6	Removed Function: Open	3

1 About This Document

This document describes the changes and new features in SALT version 2.6 (released with the first release of Dyalog version 15.0) compared with SALT version 2.42 (released with the first release of Dyalog version 14.1).

1.1 Compatibility with Dyalog Versions

SALT version 2.6 is compatible with all supported versions of Dyalog (that is, 15.0, 14.1 and 14.0).

The user command `]UUpdate` can be called to upgrade SALT to the latest version if you are running Dyalog with administrator rights.

2 Functional Changes

This chapter details the changes made to SALT for version 2.6.

2.1 Updates to Values

Calls to SALT functions that change a value now return the previous setting for that value.

2.2 Modifier Values that Include Space Characters

Modifier values that include space characters must now be enclosed within single or double quotation marks.

2.2.1 Multiple Modifier Values

Two modifiers used by SALT functions can have multiple values specified:

- The `Compare` function's `-version` modifier
- The `Snap` function's `-class` modifier

Prior to this release, multiple values were specified for these modifiers using a space character as the separator, for example, `-version=3 7`. The change to the parser has meant that this syntax has needed to be amended. Multiple values are now specified using a comma as the separator, for example, `-version=3,7`.

2.3 Performance of the Snap Function's Load File

Specifying the `Snap` function's `-loadfn` modifier generates a `<load_ws>` function that, when executed, redefines every APL object in the current workspace and runs the `⌈LX` for the workspace. The location in which to store the `<load_ws>` function can be changed from the default with a `path` modifier value.

A further modifier value, `+`, can now be included (with or without the `path` modifier value):

- the `+` modifier value should be included if the files include class dependencies as it enables the `<load_ws>` function to take account of those dependencies.
- the `+` modifier value should be excluded if the files do not include class dependencies, as this has a significant improvement on performance when the `<load_ws>` function loads multiple files from the same directory.

The syntax for the `-loadfn` modifier is now `-loadfn[=[+]path]`

2.4 Object Name Changes

The name of a SALTed object can be changed while editing it and then the object saved. Prior to this release, doing this saved the object in the original file; this could lead to confusion if the original object name was reflected in the file name. This has now changed – if the name of a SALTed object is changed while editing it, then it is not saved to the original file. Instead, the]Save function must be used to save it to a new file.

2.5 New Function: Clean

SALT maintains the links between SALTed objects and their external source files using tags. A new function, `Clean`, can be used to remove the tag information from an object (something that should not be done manually), breaking its link with all external source files.

2.6 Removed Function: Open

The `Open` function has been removed.

The functionality provided by the `Open` function remains available on the Microsoft Windows operating system by calling the `]Open` user command.