



The tool of thought for expert programming

---

Dyalog™ for Windows

# **SQAPL Release Notes**

## **Version 6.0**

Dyalog Limited

South Barn  
Minchens Court  
Minchens Lane  
Bramley  
Hampshire  
RG26 5BH  
United Kingdom

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

Dyalog is a trademark of Dyalog Limited  
Copyright © 1982-2007



Copyright © 1982-2008 by Dyalog Limited.

All rights reserved.

Version 6.0

**First Edition February 2008**

No part of this publication may be reproduced in any form by any means without the prior written permission of Dyalog Limited, South Barn, 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.

**TRADEMARKS:**

Intel, 386 and 486 are registered trademarks of Intel Corporation.

IBM is a registered trademark of International Business Machines Corporation.

Microsoft, MS and MS-DOS are registered trademarks of Microsoft Corporation.

POSTSCRIPT is a registered trademark of Adobe Systems, Inc.

SQAPL is copyright of Insight Systems ApS.

The Dyalog APL True Type font is the copyright of Adrian Smith.

TrueType is a registered trademark of Apple Computer, Inc.

UNIX is a trademark of X/Open Ltd.

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

All other trademarks and copyrights are acknowledged.

# SQAPL Release Notes

## Introduction

SQAPL Version 6.0 includes support for the following Unicode ODBC datatypes:

Data Type	SQAPL Code	Description
WCHAR	U	Wide (Unicode) fixed-length character string
WVARCHAR	W	Wide (Unicode) variable-length character string
WLONGVARCHAR	Q	Wide (Unicode) unlimited-length character string

Note: WLONGVARCHAR is identical to WVARCHAR except that you do not have to specify a maximum length when you create columns of this type. However you must specify a maximum buffer length at bind time (SQAPPrepare).

All three ODBC data types are mapped directly to the SQAPL data type C (Char). However, for some ODBC drives, it may be necessary to specify the exact ODBC data type at bind time.

## Example

The following example illustrates the use of the new data types with the SQL ODBC driver, connected to a Microsoft SQL Server Database. Please note that APL statements have been split over several lines for visual clarity.

The example is a basic Greek/English dictionary in which each record contains 4 fields:

1. **Type:** the type of word (Noun, Verb, Adverb, etc.)
2. **Greek:** the Greek word
3. **English:** the English translation
4. **Notes:** explanation

Connect to the database using the ODBC data source named SQAPLV6Test

```
SQA.Init''
sink←SQA.Connect'C1' 'SQAPLV6Test' mypasswd myuserid
```

Create a new table named Lexico containing 4 columns.

**Type** is declared as a fixed-length (12) character (ASCII) column. The ODBC datatype is CHAR.

**Greek** is declared as a variable length wide character (Unicode) column with a maximum length of 32 characters. The ODBC datatype is WVARCHAR, but the SQL Server name is NVARCHAR.

**English** is declared as a variable length character (ASCII) column with a maximum length of 32 characters. The ODBC data type is VARCHAR.

**Notes** is declared as an unlimited variable length wide character (Unicode) column. The ODBC datatype is WLONGVARCHAR, but the SQL Server name is NTEXT.

```
sink←SQA.Do'C1'('Create table Lexico
                (Type    char(12),
                 Greek   nvarchar(32),
                 English varchar(32),
                 Notes   ntext)')
```

Function `AddWord` will be used to add a record to the table.

```
▽ r←con AddWord(type eng grk notes)
[1] r←SQA.Do con('Insert into Lexico
                (Type, Greek, English, Notes)
                values (:Type<C4:,
                       :Greek<C32:,
                       :English<C32:,
                       :Notes<C128(Q):)')
                type eng grk notes
▽
```

Add 4 words ταβέρνα (taverna), εστιατόριο (restaurant), τρώω (to eat), τείζω (to feed).

```
'C1'AddWord'Noun' 'ταβέρνα' 'taverna'
      'Basic Greek eatery, often open all day,
      like a French Bistro'

'C1'AddWord'Noun' 'εστιατόριο' 'restaurant'
      'classier, typically more expensive than a
      ταβέρνα, evenings only'

'C1'AddWord'Verb' 'τρώω' 'eat'
      'Future: θα φάω, Past: έφαγα'
```

```
'C1'AddWord'Verb' 'ταΐζω' 'feed'
'e.g. to feed an animal or a baby'
```

Display the entire contents of the table:

```
3=>SQA.Do'C1' 'select * from Lexico'
Noun  ταβέρνα      taverna      Basic Greek eatery,..
Noun  εστιατόριο  restaurant  classier, typically ..
Verb  τρώω        eat         Future: θα φάω, Past: ..
Verb  ταΐζω      feed        e.g. to feed an animal ..
```

Note that SQAPL Version 6 also supports the use of Unicode characters in SQL expressions.

What is the english word for “εστιατόριο” ?

```
3=>SQA.Do'C1' 'select English from Lexico
              where Greek = :gr:' 'εστιατόριο'
restaurant
```

Which Greek words contain the string “τα” ?

```
3=>SQA.Do'C1' 'select Greek, English from Lexico
              where Greek LIKE :gr:' '%τα%'
ταβέρνα  taverna
ταΐζω    feed
```

Note that if you attempt to store Unicode data in a non-Unicode column, characters whose Unicode code points are >255 will be replaced by question marks (?).

```
sink←'C1'AddWord'Noun' 'ο μέσος όρος' '(+/ω)÷ρω'
'average or mean'

3=>SQA.Do'C1' 'select * from Lexico
              where Greek = :gr:' 'ο μέσος όρος'
Noun  ο μέσος όρος  (+/?)÷??  average or mean
```

In this example, the APL symbols  $\omega$  and  $\rho$  are replaced by “?”.

