



New Features

Contents

1	Executive Summary.....	4
1.1	Introduction.....	4
1.2	Security.....	4
1.3	Integration.....	4
1.4	Administration.....	4
1.5	Improved Usability for Development Teams.....	5
2	Security Enhancements.....	6
2.1	Multi-Factor Authentication.....	6
2.2	Entra Single Sign-on Integration.....	6
2.2.1	SSO as Sole Login Mechanism.....	6
2.2.2	Authenticating the Equinox User when SSO is Enabled.....	7
2.3	Windows Authorisation Enhancements.....	7
2.4	OAuth 2.0 Support.....	8
2.5	Buffer Overflow Prevention.....	8
3	Integration.....	9
3.1	IPv6.....	9
3.2	EqClient.dll.....	9
3.3	EqChangeLog.....	9
3.4	Merging Applications.....	10
3.4.1	UpgradeNameMatch=2.....	10
3.4.2	UpgradeNameMatch=4.....	10
3.5	DDE Integration.....	10
4	Administrator Enhancements.....	13
4.1	User Removal and Lockout from Running Applications.....	13
4.2	Progressive Search in User Management.....	13
5	Application Version Control.....	14
6	Method Language Enhancements.....	15
6.1	Method Language Editor Dark Mode.....	15
6.2	Set-Based Record Processing.....	16
6.2.1	SetIndex.....	17
6.2.2	Iterate Statement.....	17
6.3	Controlling File Name for PDF Print Output.....	18
6.3.1	SetPrintFileName.....	18
6.3.2	PrintFileName.....	19
6.4	New ways to run Method Language Code.....	19

6.5	Preselect Multiple Items on Pick and Picklist Dialogs	19
6.6	CapsCase(StringItem)	21

1 Executive Summary

1.1 Introduction

Equinox 8 delivers a range of enhancements for developers, administrators and end-users. The key benefits of Equinox 8 include:

- Enhanced security, providing greater resilience from current and future cyber-threats, and better enabling compliance with security certification requirements, such as Cyber Essentials and ISO 27001
- Improved integration with Microsoft and third-party hardware, software and standards
- Enhanced administrator tools
- Additional functionality for development teams

The following sub-sections summarise the main areas of functionality.

1.2 Security

Over the past few years, cyber security has become a major concern for organisations of all sizes, with successful cyber-attacks making the news on a regular basis.

The need for robust security has never been greater. Accordingly, Compsoft Ltd has undertaken a major investment in both technology and organisational policies, achieving accreditation to ISO 27001:2022, as well as Cyber Essentials Plus.

Equinox 8 incorporates a range of enhancements designed to increase security and to support current and future compatibility with industry security standards. Functionality includes:

- Support for Multi-Factor Authentication (MFA) for improved login security
- Enhanced support for Single Sign-On with Entra/Active Directory, reducing the burden on both users and administrators
- The ability to integrate with third-party services using the latest OAuth 2.0 protocol, which is increasingly a requirement for applications such as Microsoft Exchange
- Improved resilience against “buffer overflow” attacks

Section 2 of this document provides further details of Equinox 8 security.

1.3 Integration

Equinox 8 provides enhanced features to enable developers to integrate with third-party applications and services, other Equinox apps, and code written in other development environments, including:

- Support for IPV6
- An API to enable any application which can call Windows DLLs (including other Equinox apps) to securely log into an Equinox application, execute code, and send and retrieve data
- Enhanced capabilities for mirroring to third-party databases
- Improved support for merging separate Equinox applications
- Updated DDE support for integration with Microsoft applications such as Excel

Further details on the above functionality are provided in section 3 of this document.

1.4 Administration

Equinox 8 provides enhancements for administrators, including:

- The ability to make changes to applications while they are running, including:
 - The option to log users out from a running application
 - The ability to prevent users logging in to that application
- A progressive search in user management, to simplify the identification of users in large applications

Further details on the Admin functionality of Equinox 8 is provided in section 4 of this document.

1.5 Improved Usability for Development Teams

Equinox 8 provides the following enhancements for developers and development teams:

- Version Control, to enable team leaders and project managers to audit source code changes to applications prior to release
- Method language enhancements, including:
 - A 'dark mode', for developers who prefer a light-on-dark development interface
 - Set-Based Record Processing, substantially reducing the amount of code which needs to be written in order to identify a sub-set of system records for processing, while also reducing processing times
 - The ability to programmatically control the file name for PDF report generation
 - Additional options to trigger the auto-execution of method language procedures
 - An option for the developer to specify items to be automatically selected when generating a pick or picklist dialog
 - A new function which converts the first letter of each word to uppercase and all other letters to lowercase

The above functionality is described in detail in sections 5 and 6 of this document.

The following sections provide details of the functions available in Equinox 8. For further information on how to implement any of the below features, please contact support.

2 Security Enhancements

Over the past few years successful cyber-attacks have increased substantially, affecting organisations of all sizes, with many of the larger ones making the national and international news. Accordingly, cyber-security has become a key area of concern to customers, and a major focus of development effort for Equinox 8.

Equinox 8 incorporates a range of enhancements designed to increase security and to support current and future compatibility with industry security standards. Functionality includes:

2.1 Multi-Factor Authentication

Equinox 8 enables multi-factor authentication (MFA), enhancing security, while also enabling compliance with accreditations such as ISO27001 and Cyber Essentials.

The initial release version of Equinox 8 supports Cisco's [Duo MFA](#) solution however other industry leading MFA providers will be supported in future. If you are interested in using the MFA function with a solution other than Duo, please contact support to discuss.

If the MFA option is enabled, on entering the correct username and password, the user will be prompted to authenticate themselves via the Duo MFA app on their phone, for example by push notification or entry of a six-digit code.

Please note: MFA may be enabled for login to the Windows-based Equinox interface – Equinox for web does not support MFA, as alternative MFA solutions already exist. Likewise, where single sign-on is enabled, it is assumed that MFA will be handled prior to authentication with Equinox.

2.2 Entra Single Sign-on Integration

Equinox 7 supports single sign-on (SSO) with Active Directory. This functionality has been enhanced in Equinox 8 to enable greater security and integration with the latest Microsoft Entra software:

2.2.1 SSO as Sole Login Mechanism

This enhanced security provides the developer with greater control over access to the application in environments where SSO is to be used as the exclusive login mechanism, as follows:

- When SSO has been specified as the sole login mechanism, Equinox will no longer show the login prompt if sign-in has failed
- In Equinox 7, if the password strength was changed, Equinox forced a password change on all users, which led to unrecoverable access problems if Windows Signon or Windows Authorisation was selected. This functionality has been enhanced as follows:
 - The *force password change* flag is automatically cleared for all users if the password strength is changed to Windows authorisation or Windows Signon, or if the new "force password change now" toggle is selected and the password strength is set to Windows Authorisation or Windows Signon
 - The force password change flag is set for all users if the password strength is changed to one of the first three options in Software Settings (Case Insensitive; Case Sensitive;

- Include Numbers) AND the force password change toggle is selected, otherwise no change is made to the users
- Note that if you change the password strength and don't force a password change, the users may have to guess which case they used for their password or what their Equinox user password was, depending on which password strength option was selected
- The Equinox *change password* main menu option is greyed out if the user has been authorised using Windows Signon or Windows Authorisation
- All the options on the User tab of the Software Settings dialog are greyed out if the user does not have user management permissions
- If either the client ini key `EquinoxAuthentication=1` or the command line parameter `eqlogin` are present and the user password strength in Software Settings is set to Windows Signon or Windows Authorisation, that setting is ignored and the user is authenticated against the details held in the Equinox system database

2.2.2 Authenticating the Equinox User when SSO is Enabled

The *equinox* user is always authenticated against the system database, so an admin method to trigger the login dialog is required even when SSO is the sole login mechanism for users. A new command tail option – `eqlogin` - has therefore been added to allow the *equinox* user to override the login method and login with username and password.

2.3 Windows Authorisation Enhancements

In Equinox 7, Windows Authorisation may be used as an alternative to SSO. This does not provide SSO, but does reduce admin overhead by enabling users to use a single Windows password for login to Equinox apps as well as other systems.

In Equinox 8, in order to give administrators more control over user lockout via the method language, an optional extra parameter to the `UserResetLockout` statement is available: `UserResetLockout loginname [, flags]` which may take the following values:

-1 (or missing)	Reset user lockout, Windows authorisation unchanged
0	Reset lockout and disable Windows authorisation only
1	Set the user lockout flag and disable Windows authorisation only
2	Reset lockout and enable Windows authorisation only
3	Set the user lockout flag and enable Windows authorisation only

It is also now possible to programmatically detect whether the application has been set to use Windows authorisation or SSO using the following syntax:

```
UserDetails loginname [, fullname, usergroup, category, userexpirydate, passwordexpirydate, lastaccessdate, flags]
```

The final parameter of `UserDetails` now has 4 added to its value if Windows authorisation only is enabled.

2.4 OAuth 2.0 Support

Equinox 8 enables integration with third-party services using the latest OAuth 2.0 protocol, which is increasingly a requirement for integration with applications such as Microsoft Exchange. Please note: integration also requires the latest EQSmail DLL.

Microsoft is deprecating [basic authentication](#) for Exchange Online, which includes the use of app passwords for POP, IMAP, and SMTP AUTH, with the final removal of SMTP AUTH scheduled to take place in 2027.

2.5 Buffer Overflow Prevention

Equinox 8 incorporates additional security to prevent malicious actors from causing buffer overflow by inserting data into the Equinet.ini file.

3 Integration

Equinox 8 provides enhanced features to enable developers to integrate with third-party applications and services, other Equinox apps, and code written in other development environments:

3.1 IPv6

IPv6 support has been added to the Equinox 8 client, server and webserver.

In order to use IPv6, add the ini key `UseIPv6=1` to the relevant ini files. Please note: you must also ensure you are using hostnames rather than numeric IP addresses in your connection string.

3.2 EqClient.dll

This feature allows any application which can call Windows DLLs to log into an Equinox application, execute code, and send and retrieve data. It can be used from anywhere that you can call DLLs - possible uses include:

- Communicating with an Equinox application from one or more other Equinox applications
- Communicating with an Equinox application from another executable running on the same machine, which might be written using a different technology, e.g. .NET, Java, scripting languages etc.

EqClient has the following functionality:

- Login
- Logout
- Execute method (a pre-existing Equinox application method)
- Execute macro (text containing executable method language code)
- Get workarea value
- Set workarea value

3.3 EqChangeLog

EqChangeLog enables developers to write their own mirror interfaces to third-party databases, for example those not supported out of the box by Equinox.

The EqChangeLog mirror DLL can be configured to use the MRC delayed update system or, alternatively, to run as a "direct" mirror, where the change log is updated inline by the server, along with the Equinox data updates. In order to do this the system should be configured as follows:

- All relevant databases should be converted to use EqChangeLog.dll
- The following keys are required in client and single user ini files:
 - `MirrorUpdateMode=3`
 - `<appname>MirrorMode=3`

An HDR table description and a single log stream are produced under the folder `\Database\EqChangeLog`.

3.4 Merging Applications

3.4.1 UpgradeNameMatch=2

Enhancements have been made to the process of merging applications when using `UpgradeNameMatch=2` (to merge items into the currently selected dictionary within the destination application).

The following provisos and limitations should be taken into consideration:

- Dictionaries should be included in the pub file, as they help with the name matching, but are ignored. No dictionary cross-references are imported and must be added manually.
- Unmatched definitions and workareas are renumbered to be compatible with the target application
- Workarea values are not imported
- Public procedures are imported as in normal name match upgrades
- Other parameter files are imported and associated with the current dictionary
- It is advisable to run the check application option on the System menu, and do a global resave
- The application should be backed up prior to merge so it can be restored if a problem occurs

The entire folder structure is now published for pictures and glyphs, not just the root folder.

There are a large number of potential scenarios in which this function may be used, so developers are encouraged to try it out and report any issues to Compssoft Support for resolution.

3.4.2 UpgradeNameMatch=4

When applying an upgrade to an unpublished application, Equinox would normally do a name match upgrade and ignore the `UpgradeNameMatch` ini key. However if the target application is the same unpublished application, it may be helpful to apply a logical match upgrade. It's now possible to override this automatic behaviour by adding 4 to the desired value of the `UpgradeNameMatch` ini key. Thus for example, `UpgradeNameMatch=6` will apply a logical name match **and** merge the dictionary.

3.5 DDE Integration

Equinox 8 provides a new version of the DDE DLLs which avoids a problem call to establish DDE communication with Excel. The old version relies on sending a message to every program with a window in the system, in order to find one which can deal with the request being made. However if any one of those programs is not processing Windows messages for their window, it simply hangs waiting for a response. The new DLL's allow the developer to specify the name of the window the application is communicating with. Two main changes are required in the application method, as follows:

- The code that waits for Excel to load and then finds its window – see below
- The call to the `dde_initiate` (which still exists) is replaced by call to `dde_initiate_direct`, which takes the extra parameter of the Excel window that it thinks it should use.

Example:

```
external "DDE" dde_initiate handle, readonly ptr stringz, readonly ptr  
stringz returns handle
```

becomes:

```
external "DDE" dde_initiate_direct handle, readonly ptr stringz, readonly  
ptr stringz, handle returns handle
```

New External call:

```
external "user" FindWindow readonly ptr stringz, readonly ptr stringz  
returns handle
```

```
string sheetname, title, vsFilePath
```

```
handle hndexcel
```

```
stringz vszitem, vszdata, com, service
```

```
service = "Excel"
```

```
sheetname = "DDE_Test.xlsx"
```

```
title = "DDE_Test - Excel"
```

```
execute task vsfilePath & "\excel.exe"
```

```
for i = 1 to 10
```

```
    Sleep 1
```

```
    FindWindow nullptr, title, hndexcel
```

```
    if hndexcel then exit for
```

```
next
```

```
if not hndexcel then
```

```
    print #"Can't find Excel window with title "; title; ""
```

```
    i = 1
```

```
    title = "DDE_Test.xlsx - Excel"
```

```
    for i = 1 to 10
```

```
        Sleep 1
```

```
        FindWindow nullptr, title, hndexcel
```

```
        if hndexcel then exit for
```

```
    next
```

```
end if
```

```
print "hndexcel = ", hndexcel
```

```
if not hndexcel then
```

Equinox 8 New Features

```
        print #"Can't find Excel window with title "; title; ""
        exit method
    end if

    |      Open communications channel with Excel
dde_initiate_direct Hwnd, service, vszdata, hndexcel, hExcel

    |      Test if service responded
if hnd = 0 then
    print #"Unable to communicate with Excel."
    exit method
end if
```

The above code replaces this old line:

```
dde_initiate Hwnd,"Excel","System", hExcel
```

All other DDE code from this point on remains as it was.

4 Administrator Enhancements

4.1 User Removal and Lockout from Running Applications

In previous versions of Equinox, it was not practical to make administrative changes to a running application. This was because a) there was no way to remove users without adverse consequences and b) there was no way to prevent users logging in, as this would create conflicts between the administrative changes and user changes.

Equinox 8 provides a new Maintenance Mode, which enables users to be locked out without halting the application. The solution enables the administrator to log out all users at once - to warn the users, admins may write an application method to send a system-wide message.

The `LockApplication` statement is used to remove any users who have not logged out following the system-wide message. `LockApplication` is an existing Equinox statement which has been enhanced in Equinox 8 to take an optional numeric parameter - if it is set to 2, this triggers the functionality described above. To reverse the process, call `UnlockApplication`.

In addition, users trying to log into to an application in Maintenance Mode may be notified that the application is temporarily unavailable, while not interfering with the maintenance operations.

4.2 Progressive Search in User Management

Equinox User Management has a progressive find on the user list. As with other progressive finds in Equinox, this allows the user to type the required name and the list will jump to the closest match, making it much easier to find a user in a system with a large number of defined users.

5 Application Version Control

In Equinox development teams, team leaders or project managers may need to audit source code changes to the application prior to release. Equinox 8 incorporates the following features to enable this:

- Save Audit Enable allows developers to audit the changes made in a publish operation. The Software Settings audit tab now contains an extra checkbox entitled "Enable software audit"; this checkbox is only enabled if the user has enable/disable auditing permission. If enabled, every time a module is saved, the module's publish flag is unselected for the next publish operation.
- An option to "*View non-upgrade items*" is available in the right-click Navigator menu, which shows the list of changed modules. The code may then be reviewed and the status set to "Include in upgrade creation". This triggers an optional `AutoexecAfterPublishEnabled` procedure, where developers may capture any details necessary. On saving the module, the existing `AutoexecAfterSaveFile` procedure is also available to audit individual module changes. The publish process final dialog now also displays the number of items included, and the number not included, in the publish file; the log also contains this information.

6 Method Language Enhancements

6.1 Method Language Editor Dark Mode

In Equinox 8, the method language editor now has support for a 'Dark Mode', for developers who prefer a light-on-dark development interface. Dark Mode can be set to work with or without syntax highlighting.

The mode can be changed in Software Settings and also can be detected in the method language via the read-only system workarea `SysDarkMode`. A subdued set of default colours has been chosen for Dark Mode.

The optional syntax colour definition system in the S/U or client ini file has been updated. In all previous builds the colour settings were controlled by ini keys of the form:

```
SyntaxColour##=<r,g,b>
```

where ## was a number between 1 and 11, and r, g and b were numbers between 0 and 255 representing the value for each colour, combined to make an 'RGB' value, or alternatively:

```
SyntaxColour##=0xRRGGBB
```

where RRGGBB represented a hexadecimal number between 0X0 and 0xFFFFF; each two-digit segment contained the value for that colour.

Additional keys are now provided in order to maintain a set of Dark Mode colours alongside the normal ones, and there is a new facility to select system colours:

```
SyntaxColour##=<r,g,b>
```

```
SyntaxColour##=0xRRGGBB
```

```
SyntaxColour##=<SystemColour>
```

```
SyntaxDarkColour##=<r,g,b>
```

```
SyntaxDarkColour##=0xRRGGBB
```

```
SyntaxDarkColour##=<SystemColour>
```

The set of colours available to be changed is now 0 to 12, as follows (1 to 11 are the same as previous versions):

- 0: background
- 1: variables, fields, workareas
- 2: reserved words
- 3: functions
- 4: external module names
- 5: predefined constants
- 6: preprocessor definitions
- 7: statements
- 8: attributes
- 9: public procedures and calls
- 10: comments
- 11: quoted strings
- 12: default text colour

The value `<SystemColour>` should be set to one of the following system colour names, which can also be found at the end of the forms colour selector, along with the colour assigned in the current user interface theme:

```
ColourScrollbar  
ColourDesktop  
ColourActiveCaption  
ColourInactiveCaption  
ColourMenu  
ColourWindow  
ColourWindowFrame  
ColourMenuText  
ColourWindowText  
ColourCaptionText  
ColourActiveBorder  
ColourInactiveBorder  
ColourAppWorkspace  
ColourHighlight  
ColourHighlightText  
ColourBtnFace  
Colour3DShadow  
ColourGreyText  
ColourBtnText  
ColourInactiveCaptionText  
Colour3DHighlight  
Colour3DDkShadow  
Colour3DLight  
ColourInfoText  
ColourInfoBk  
ColourTransparent  
ColourHotlight
```

Please note: the colours for selected text cannot be changed.

6.2 Set-Based Record Processing

Equinox 8 enables Set-Based Record Processing, substantially reducing the amount of code which needs to be written in order to identify a sub-set of system records for processing, while also significantly reducing processing times.

This incorporates changes to the `SetIndex` parameter, as well as a new Iterate function:

6.2.1 SetIndex

An extra optional limits parameter has been added to `SetIndex` and `GroupSetIndex`:

```
[Group]SetIndex indexname [, username, firstkey, lastkey, limits]
```

The limits parameter can take the following values to determine whether or not to include the first and last key values:

- 0: lower <= data <= upper (default)
- 1: lower < data <= upper
- 2: lower <= data < upper
- 3: lower < data < upper
- +4: automatic record updating (see below)

For example, if a range is specified on the alphabet from J to N, the following letters are included, depending on the value of the limits parameter:

- 0: J K L M N
- 1: K L M N
- 2: J K L M
- 3: K L M

6.2.2 Iterate Statement

A new language construct, `iterate/end iterate` can be used to replace the following typical template code:

```
subtable tablename
    SetIndex indexname, username
    FindRecord ge, firstkey
    while not SysError and currentkey <= lastkey
        [user code]
        [exit subtable]
        [continue subtable]
    NextRecord
end while
end subtable
```

The iterate version looks like this:

```
iterate tablename, indexname, username, firstkey, lastkey, limits
    [user code]
    [exit iterate]
    [continue iterate]
end iterate
```

The parameters `indexname`, `username`, `firstkey`, `lastkey` and `limits` are all optional.

The iterate block technology and `SetIndex/NextRecord` can optionally update the current record if it has changed. In order to enable updating, add 4 to the final limits parameter. Additionally, if `ReplaceRecord` is used within the block it is ignored, and the update occurs before the `NextRecord` functionality in end/continue iterate. This can improve processing speed considerably in complex record processing code which has multiple sections like the following, without having to make changes to it:

```
if [somecondition] then
    [change record data]
    ReplaceRecord
end if
```

This flag is not inherited by nested sub-tables.

Also note that code which checks for `SysError` after `ReplaceRecord` will not work correctly as `SysError` will not be set until the `NextRecord` occurs. In this situation it would be better to leave the code as it is; however If the `NextRecord` succeeds and the `ReplaceRecord` fails, `SysError` is set to 6.

The statements `Exit Iterate` and `Continue Iterate` work in the same way as other block statements. However please note that `Continue Subtable` is of limited use, whereas `Continue Iterate` passes control to the `NextRecord` functionality in `End Iterate`.

6.3 Controlling File Name for PDF Print Output

Equinox 8 provides two new method language statements, `SetPrintFileName` and `PrintFileName`:

6.3.1 SetPrintFileName

- Syntax: `SetPrintFileName [name]`
- Action: `[Statement]` sets the name of the file to be used in the next report configured for print to file
- Scope: Usable anywhere

The parameter name is optional, and if supplied should contain the full path and name of the output file. If omitted, any name is removed. Once a report has run, the current file name (if any) is removed. If an error occurs, the system workarea `SysError` is set to a non-zero value, otherwise to zero.

This example sets the printer driver specifics and an output filename, and then runs the report:

```
|   Select a device driver and print to file
ChoosePrinter 2, "Microsoft Print to PDF",,,,,,2
    if SysReply < 0 thenprint "Error", SysError, "occurred choosing the
printer"
        exit method
    end if

|   Select a location and name for the output file
path = SysAppHomeDir & "\\Reports"
MakeDirectory path
SetPrintFileName path & "\\TestReport.pdf"

|   Run the report
SysPause = 0
execute report "TestReport"
if FileExists(path & "\\TestReport.pdf") then print "Ok" else print "Report
not found"
```

6.3.2 PrintFileName

Syntax: `PrintFileName`

Action: `[Function]` returns the current print filename

Scope: Usable anywhere

This function returns the current print filename, or "" if none is set.

6.4 New ways to run Method Language Code

Additional options have been added to the list of events which will trigger auto-execution of method language procedures. These are:

- Delete parameter file – `AutoExecBeforeDeleteFile`
- Server startup – `AutoExecAfterStartup` (Server side only)
- Server shutdown – `AutoExecBeforeShutdown` (Server side only)

6.5 Preselect Multiple Items on Pick and Picklist Dialogs

Equinox 8 enables the developer to specify items to be automatically selected when generating a pick or picklist dialog. This means that, if a user has selected certain data items, for example the months June and August from a list of calendar months, when they return to the picklist, June and August will remain selected and need not be reselected.

`PickSelect [0=clear, 1=add, 2=query] default=add]`

Selects/deselects/queries the current record to be highlighted in a forthcoming Pick statement.

`PickListSelect line [,select]`

- line=0-256 (0=all)
- select: 0=clear, 1=set (default), 2=query

Selects/deselects/queries the 1-based line number to be highlighted in a forthcoming PickList statement. If 0 is supplied for line, all items are affected.

Please note: in a multi-select pick, the Return key now only selects the current item if no other selection has been made. In order to select the current line, press the space bar first and then press Return.

Example Pick:

```
number vnIndex, vnSelections
string vsselected

| Clear the selections
PickSelect 0

subtable Customer
  SetIndex CustNameIX
  FindRecord ge, "A"

  | Pre-Select first Customer
  print "Pre-selected: " & CustName
  PickSelect

  NextRecord
  NextRecord

  | Pre-Select third customer
  print "Pre-selected: " & CustName
  PickSelect

  NextRecord
  NextRecord
  NextRecord

  | Pre-Select sixth customer
  print "Pre-selected: " & CustName
  PickSelect
end subtable      | Customer

PopupTitle "Select Customer/Customers"
PopupFlags 64
PopupSize , , 40,
PickColumnWidths 20,20,20
PickFields CustName, CustAddress, CustTelephone
vsSelected = PickRange(CustName,, CustNameIX, "A", "C")

| Note no. of multiple selections
vnSelections = SysResult
print "vnSelections = ", vnSelections
vnIndex = 0

| Multiple selections ?
if vnSelections > 0 then
  print "There were <" & vnSelections & "> selections made"

  | Loop through list of selections
  for vnIndex = 1 to vnSelections
    PickSelection vnIndex, vsSelected
    print "Selection " & vnIndex & " - <" & vsselected & ">"
  next
end if
```

Example PickList:

```
number vnIndex, vnSelections
string s

PopupTitle "Select A Number"
PopupFlags 64
PopupSize , , 17

| Clear the selections
PickListSelect 0, 0

PickListSelect 2, 1
PickListSelect 5, 1

s = PickList("/One/Two/Three/Four/Five/Six/Seven/Eight", True)

| Note no. of multiple selections
vnSelections = SysResult
vnIndex = 0

| Multiple selections ?
if vnSelections > 0 then
    print "There were <" & vnSelections & "> selections made"

    | Loop through list of selections
    for vnIndex = 1 to vnSelections
        PickListSelection vnIndex, s
        print "Selection " & vnIndex & " - <" & s & ">, Original Index
- <" & SysReply & ">"
    next
end if
```

6.6 CapsCase(StringItem)

There is now a new `CapsCase(StringItem)` function, which converts the first letter of each word in `StringItem` to uppercase and all other letters to lowercase. This is similar to the `StrCase StringItem, CaseExpression` statement, when supplying a `CaseExpression` of 2.