APL64 Command Line Arguments

Table of Contents

Overview 2

Location of Command Line Arguments........ 2

Format of Command Line Arguments........ 2

Command Line Parameter Synonyms........ 3

Undefined APL64 Command Line Arguments........ 4

APL64 Command Line Arguments 4

- Apl64ConfigSrcs 4
- Apl64ConfigPath 5
- Apl32ConfigSrcs 5
- Apl32ConfigPath 6
- ShowEmbeddedResourceNames 7
- SplashImagePath........ 8
- SplashImageCenter 8
- SplashImageTop 8
- WsToLoad........ 9
- User 10
- UcmdFile 10
- UcmdInit........ 10
- APL32 11
- ActivationLicenseId 11
- ActivationLicensePassword........ 12
- ActivationProxyServerUserName 12
- ActivationProxyServerUserPassword........ 12
- ActivationProxyServerUrl........ 13
- ActivationWebProxyType........ 13
- CheckComponentFile 14
- CompFileCommitSync 14
- NavigateDirs........ 14
- EmitLongDateFormat........ 15
- PathCase 15

Editing of APL64 or APLNow32 Configuration Files 15

Overview

Command line arguments may be used to control the configuration of an instance of APL64. The configuration is established when an instance of APL64 is created. Configuration of an APL64 instance involves the separate configuration of the APL64 component and its optional APLNow32 Win32 component.

The application of potential sources of APL64 configuration information when initializing an APL64 instance is controlled by the Apl64ConfigSrcs and Apl32ConfigSrcs command line arguments.

The potential sources of APL64 configuration information are:

1. Default values for the configurable features of the APL64 product which are controlled by APLNow LLC. This APL64 product default configuration information forms the basis of the configuration of an APL64 instance. The Apl64ConfigSrcs and Apl32ConfigSrcs command line arguments control the use of the additional potential sources of configuration for the initialization of an APL64 instance.

2. If the APL64 instance is based on a runtime executable, the APL64 programmer may have provided APL64 xml-format and APLNow32 ini-format configuration files which were embedded into the Windows runtime executable or the cross-platform component version of APL64. Embedded configuration files do not apply to the APL64 developer version.

3. The APL64 programmer may provide physical APL64 xml-format and APLNow32 ini-format configuration files with paths specified in the values of the Apl64ConfigPath or Apl32ConfigPath command line arguments.

4. Configuration information provided by command line arguments, other than the Apl64ConfigPath or Apl32ConfigPath command line arguments, if any, may be used. Command line arguments are optional. The operating system environment may impose limits on the number of characters in the command line. Using an APL64 xml-format or APLNow32 ini-format configuration file in conjunction with the APL64 command line arguments ‘Apl64ConfigPath=…’ or ‘Apl32ConfigPath=’ avoids these limits.

Location of Command Line Arguments

Command line arguments are text elements associated with the APL64.exe developer version executable or an APL64 runtime executable. These executables and the optional command line arguments are usually specified in a Windows shortcut file or a Windows batch file.

Format of Command Line Arguments

- Each command line argument is separated on the command line by one or more space characters.

- A command line argument is composed of the ‘argname’ and ‘argvalue’ separated by the equals (=) glyph, i.e. argname=argvalue.
- Examples:
  - Argname1=argvalue1
  - Argname1=argvalue1 argname2=argvalue2 …
- The argname portion of a command line argument can have no spaces and is case insensitive.
- Quotation marks must enclose an argvalue which contains spaces.
- Multiple command line arguments are separated by space glyphs.
- For some command line arguments, the ‘argname=’ prefix is optional.
- Some command line arguments can be specified in the APL64 xml-format configuration file.
- When used on the command line, command line arguments which have a Boolean value can have their value specified as 1 instead of true or 0 instead of false. This option applies only to APL64 configuration values.
  - Examples
    - A64CS=F
    - For an APL32 command line argument of the format 'a32=[section]key=value', the specification of the value element must conform to the APL32 definition of that value.
- Optional command line prefix abbreviations
  - When used on the command line, some command line arguments support an abbreviated 'argname=' prefix.
  - An abbreviated command line argument prefix is not case sensitive.
  - When a command line argument can be specified in the APL64 xml-format configuration and is specified there, the non-abbreviated format is required.
  - Examples:
    - A64CS=N is equivalent to A64CS=None
    - A64CS=F is equivalent to Apl64ConfigSrcs =F
    - a32=[Config]CTL3D=1 is equivalent to apl32=[Config]CTL3D=1

Command Line Parameter Synonyms

Command Line Argument Prefix	
Full	**Abbreviation**
ActivationLicenseId	ALI
ActivationLicensePassword	ALP
ActivationProxyServerUserName	APSUN
ActivationProxyServerUserPassword	APSUP
ActivationProxyServerUrl	APSU
ActivationWebProxyType	AWPT
Apl64ConfigSrcs	A64CS
Apl64ConfigPath	A64CP
APL32	A32
Apl32ConfigSrcs	A32CS
Apl32ConfigPath	A32CP
CheckComponentFile	CCF

CompFileCommitSync	CFCS
EmitLongDateFormat	EDLF
NavigateDirs	NAVD
PathCase	PC
ShowEmbeddedResourceNames	SERNS
SplashImagePath	SIP
SplashImageCenter	SIC
SplashImageTop	SIT
User	U
UcmdFile	UCF
UcmdInit	UCI
WsToLoad	WSTL

Undefined APL64 Command Line Arguments

Command line arguments which are undefined by APL64 will be ignored during the configuration of an APL64 instance and no exception will be thrown by APL64 during the configuration of an APL64 instance. The third and tenth elements of ⎕sysinit may be used by the APL64 programmer at runtime to obtain all the APL64 command line arguments. Undefined command line arguments may be used by the APL64 programmer for application-specific purposes.

APL64 Command Line Arguments

Apl64ConfigSrcs

- Supported argvalues, with supported abbreviations in parentheses:
  - None (N)
  - FileConfig (F)
- Default value: “F”
- Argvalue is not case sensitive
- This command line argument is not saved in the APL64 xml-format configuration file, so it does not apply to future APL64 instances unless it is present as an APL64 command line argument.
- Ordered use of APL64 configuration information:
  - None (N): APL64 product defaults used. Runtime-embedded configuration file, physical configuration file, if any, is deleted and command line arguments are not used.
  - FileConfig (F): APL64 product defaults used, overridden by a runtime-embedded configuration file, overridden by a physical configuration file and finally overridden by any command line arguments.
- Examples:
  - Apl64ConfigSrcs=None
  - A64CS=N
  - A64CS=F
  - Apl64ConfigSrcs=FileConfig

Apl64ConfigPath

- Supported argvalue: Full path to the APL64 xml-format configuration file.
- During the initialization of an APL64 instance, the file specified by the Apl64ConfigPath command line argument may be accessed to obtain the configuration information contained within it
- Since APL64 will read this file during the configuration of an APL64 instance and may write to this file during an APL64 instance, the user-specified path should provide the necessary workstation permission for those actions.
- This command line argument is not saved in the APL64 xml-format configuration file, so it does not apply to future APL64 instances unless it is present as an APL64 command line argument.
- The format and case-sensitivity of the value depends on the operating system environment.
- The ‘Apl64ConfigPath=’ prefix is optional for this command line parameter, however, if the prefix is omitted, the ‘.xml’ file name extension is required.
- If there is no Apl64ConfigPath command line argument or if the initial value of an existing Apl64ConfigPath command line argument in not a filename of a file accessible to APL64, the ‘default location’ for configuration files is checked for a file with the same file name as the APL developer or runtime version executable, but with the file extension ‘.xml’.
- To assure adequate read/write file permissions for the APL64 xml-format configuration file, the ‘default location’ is the path provided by the ⎕userpath system function.
- Example for the Windows environment:
    - Apl64ConfigPath=”c:\myapl64 configpath\apl64config.xml”
    - “c:\myapl64 configpath\apl64config.xml”
    - Apl64ConfigPath=”c:\myapl64 configpath\myapl64config.myextn”
    - A64CP=”c:\myapl64 configpath\myapl64config.myextn”

Apl32ConfigSrcs

- Supported argvalues, with supported abbreviations in parentheses:
    - None (N)
    - FileConfig (F)
- Default value: “F”
- Argvalue is not case sensitive
- This command line argument is not saved in the APL32 ini-format configuration file, so it does not apply to future APL64 instances unless it is present as an APL64 command line argument.
- This command line argument applies only to the Windows operating system environment because it pertains to the Win32-dependent components of APL64.
- Ordered use of APLNow32 configuration information:
    - None (N): APLNow32 product defaults used. Runtime-embedded configuration file, if any, is not used, physical configuration file, if any, is not used and other command line arguments, if any, are not used.
    - FileConfig (F): APLNow32 product defaults used, overridden by an APLNow32.ini configuration file embedded in and materialized by an APL64 Windows runtime executable, overridden by a physical configuration file, if any, and finally overridden by any command line arguments.
- Examples:

  - Apl32ConfigSrcs=None
  - A32CS=N
  - A32cS=F
  - APL32ConfigSrcs=F

Apl32ConfigPath

- Supported argvalue: Full path to the APLNow32 ini-format configuration file, if any.
- The ‘Apl32ConfigPath=’ prefix is optional for this command line parameter, however in this case the ‘.ini’ file name extension is required.
- This command line argument applies only to the Windows operating system environment because it pertains to the configuration of the Win32-dependent components of APL64. The value of this command line argument is not case-sensitive.
- When an instance of APL64 is started, depending on the existence and value of the Apl32ConfigSrcs command line argument, other APL32 command line arguments, if present, may override the configuration information in this APL32CONFIGPath file.
- Upon the use of a feature of APL64 which requires the Win32-dependent components of APL64, this APL32 ini-format configuration file may be accessed to obtain the APLNow32 start-up state.
- Since during an APL64 instance this file will be read and may be written, the user-specified path should provide the necessary workstation permission for those actions.
- This command line argument is not saved in the APL64 xml-format configuration file, so it does not apply to future APL64 instances unless it is present as an APL64 command line argument.
- The Win32-dependent features of APL64, such as ⎕wi, are supported by APLNow32 components:
  - The required APLNow32 executable file named “APLNow32.exe”
    - If the APL64 instance is the developer version of APL64, this file is installed to the same folder as the APL64.exe component
    - If the APL64 is a runtime instance of APL64, this file is embedded in the Windows Runtime Executable and materialized at runtime to the same folder as the APL64.dll component
    - In APL64 the file name ‘APLNow32.exe’ is fixed and cannot be modified by the APL64 programmer
  - The optional APLNow32 adf-format file named “APLNow32.adf” file, if required, must in the same folder as the APLNow32.exe executable file.
  - The optional APLNow32 ini-format configuration file which can be located and named using the Apl32ConfigPath command line argument. The APLNow32.exe component may initiate as search for this file or provide default values if it is not found in the provided location.
- Default APL32ConfigPath value:
  - If this command line argument is not present, the default value is “”. In this case APLNow32.exe will use an APLNow32.ini file if it exists in the same folder as APLNow32.exe
  - If the Apl32ConfigSrcs=N or A32CS=N command line argument is present:
    - The APLNow32.exe ⎕wcall 'W_IniFile' system function will return a special non-existent, randomly-generated ini filename so that APLNow32.exe will not search for any other ini file, for example:

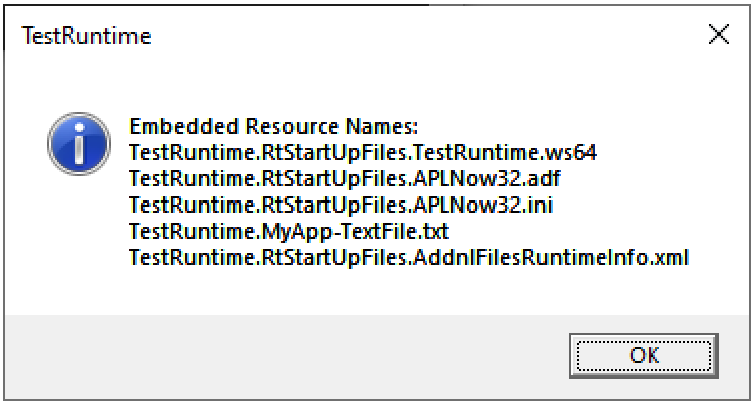
…\APLNow32-71936A7A-75DB-498A-9D08-83AD102A2DB2.ini

  - The APLNow32.exe ⎕wcall 'W_Ini' system function may be used during the current instance of APLNow32.exe to read and write to the current ini file, creating it if it does not exist if the user credentials for its path are suitable. For subsequent instances of APLNow32.exe, if the Apl32ConfigSrcs=N or A32CS=N command line argument is present, the ini file associated with the prior instance of APLNow32.exe will be ignored.

- Examples for the Windows environment:
  - Apl32ConfigPath=”c:\myapl32 configpath\APLNow32.ini”
  - A32cp=”c:\myapl32 configpath\APLNow32.ini”.
  - ”c:\myapl32 configpath\APLNow32.ini” in this case the ‘A32CP’ or ‘APL32CONFIGHPATH’ command line argument prefix is not used, so the .ini file name extension is required.

ShowEmbeddedResourceNames

- This command line argument applies only in the APL64 runtime environment and should be removed when an APL64 runtime application is deployed in a production environment.
- An APL64 runtime executable contains files required for the APL64-based runtime application as ‘embedded resources’. This feature enables an APL64 programmer to create a single file Windows exe-format executable containing all the necessary components for the application to be deployed and run on an end user’s target workstation.
- After creating an APL64 runtime executable using the utility provided in the APL64 developer version, it is often convenient to check that the proper files have been included as embedded resources.
- Run the APL64 runtime executable in a Windows batch file or shortcut with this command line argument.
- Supported argvalue: True or False
- Default value: False
- Argvalue is not case sensitive
- This command line argument is not saved in the APL64 xml-format configuration file.
- When this command line argument is present with a value of true, a message box will be displayed containing the embedded resource names in the APL64 runtime executable. The APL64R.dll and APLNow32.exe modules will not be listed as embedded resources, although they are included in an APL64 runtime executable.



Embedded Resource Names:
TestRuntime.RtStartUpFiles.TestRuntime.ws64
TestRuntime.RtStartUpFiles.APLNow32.adf
TestRuntime.RtStartUpFiles.APLNow32.ini
TestRuntime.MyApp-TextFile.txt
TestRuntime.RtStartUpFiles.AddnlFilesRuntimeInfo.xml

- Examples:
    - SERNS=1
    - ShowEmbeddedResourceNames=true

SplashImagePath

- The full path of the optional splash image file to be displayed when the developer or run-time version of APL64 is started.
- This information is saved in the APL64 xml-format configuration file.
- Supported image file formats include bmp, gif (static and animated), jpeg, png.
- If the non-empty, SplashImagePath value is not the filename of a file accessible to APL64, no exception will be thrown, no splash image will be presented and the action of the □splash system function will be disabled during the current APL64 instance.
- The display of a splash image will be ended automatically when the APL64 instance is fully initialized, the WsToLoad, if any, has been loaded and the □lx function, if any, has completed execution. Use □splash 0 to end the display of a splash image at an earlier time.
- APL64 Developer Session:
    - If the SplashImagePath is empty or not present, the APL64 default splash image will be presented during the initialization of an APL64 developer session instance.
    - To suppress the display of the APL64 default splash image, provide a non-empty SplashImagePath value of a file which does not exist.
- APL64 Runtime Executable:
    - Only an APL programmer-provided splash image will be displayed.
- Examples for the Windows environment:
    - SplashImagePath="c:\my splash folder\app1 splash.gif"
    - sip="c:\my splash folder\app1 splash.gif"
    - SplashImagePath=HTTP://www.myAppWebsite/appsplash.png

SplashImageCenter

- Supported argvalue: True or False
- Default value: False
- Argvalue is not case sensitive
- This information is saved in the APL64 xml-format configuration file.
- When argvalue is true, the splash image, if any, will be centered on the ensemble of all displays connected to the workstation.
- When argvalue is false, the splash image, if any, will be centered on the operating-system-designated 'primary' display.
- Examples:
    - SplashImageCenter=true
    - SIC=1

SplashImageTop

- Supported argvalue: True or False
- Default value: False

- Argvalue is not case sensitive
- This information is saved in the APL64 xml-format configuration file.
- When argvalue is true, the splash image, if any, will be the ‘topmost’ window.
- Examples:
  - SplashImageTop=true
  - SIT=1

WsToLoad

- During the configuration of an APL64 instance an attempt will be made to load this workspace and execute the APL executable statement in the ⎕lx system variable value of the workspace, if present.
- Supported argvalues:
  - Full path to the APL64 workspace to load *with* the file name extension ‘.ws64’. This option is available when the ‘WsToLoad=’ or ‘WSTL=’ command line argument prefix is present or not present.
  - Full path to the APL64 workspace to load *without* the ‘.ws64’ file name extension. This option is only available when the ‘WsToLoad=’ or ‘WSTL=’ command line argument prefix is present. In this case the file name extension, ‘.ws64’ (case sensitive), will be used when an attempt to load the workspace is made. The physical APL64 workspace file referenced in this case must have the ‘.ws64’ file name extension.
  - Full path to the APL+Win workspace to load *with* the ‘.w3’ file name extension. This option is available when the ‘WsToLoad=’ or ‘WSTL=’ command line argument prefix is present or not present.
- This command line argument is not saved in the APL64 xml-format configuration file, so it does not apply to future APL64 instances unless it is present as an APL64 command line argument.
- The format and case-sensitivity of the argvalue depends on the operating system environment.
- The ‘WsToLoad=’ or ‘WSTL=’ prefix is optional for this command line parameter. When the prefix is not present, the ‘.w3’ or ‘.w64’ file name extension is required.
- The ‘WsToLoad=’ command line argument is available only the APL64 developer version. For the APL64 runtime version:
  - The required runtime-ready workspace to load is provided by the APL64 programmer when an APL64 Windows Runtime Executable is created.
  - The optional runtime-ready workspace to load is provided by the APL64 programmer when an APL64 Cross-platform Component is created.
- Examples for the Windows environment:
  - WsToLoad=”c:\my ws to load\myapp”
  - wstl=”c:\my ws to load\myapp”
  - WsToLoad=”c:\my ws to load\myapp.ws64”
  - ”c:\my ws to load\myapp.ws64”
  - WsToLoad=”c:\my ws to load\myWin32App.w3”
  - WsTL=”c:\my ws to load\myWin32App.w3”
  - ”c:\my ws to load\myWin32App.w3”

User

- The numeric value specified by the argvalue will be used as the APL64 □ai[1] value during the work session.
- This information is not saved in the APL64 xml-format configuration file.
- The default value is 1.
- The default value is used when a new instance of APL64 is created, and there is no valid User= command line argument.
- Examples:
  - User=1
  - u=22

UcmdFile

- Supported argvalue: The full path of the 'main' user commands definition file, e.g. …\UCMDS.SF.
- The specified user command file must be internally formatted as an APL64 component file.
- This information is saved in the APL64 xml-format configuration file.
- The format and case-sensitivity of the argvalue depends on the operating system environment.
- If the UcmdFile command line argument or the APL64 xml-format configuration file does not specify the this path, the default value is 'Combine'□path □userpath 'UCMDS.sf'.
- In an APL64 Developer version instance this path is the value of the APL64 system function □ucmdfile.
- Examples for the Windows environment:
  - Ucf="c:\my folder\UCMDS.SF"
  - UCMDFile="c:\my folder\UCMDS.SF"

UcmdInit

- Perform the initialize user command files: 0/False 1/True
- APL64 interpreter action when true in an APL64 Developer version instance:
  - The APL64 default user command files are copied (with overwrite) to the folder specified by the value of the APL64 system function □ucmdfile and named: "UCMDSW.sf", "UCMDS2.sf", "UCMDS3.sf", and "UCMDS.sf".
  - Component #4 of the UCMDS.sf is updated to provide the appropriate folder paths.
  - Set the UcmdInit value to false so that for subsequent APL64 instances, the initialization is not performed.
- For a new installation of the APL64 Developer version the value of UcmdInit is true.
- This command line argument does not apply to the APL64 runtime version.
- This information is saved in the APL64 xml-format configuration file.
- Examples:
  - UcmdInit=true
  - UcmdInit=0
  - UCI=1
  - UCI=false

APL32

Settings to be used for configuration of the APLNow32 components of APL64 use the ‘APL32=’ prefix and the appropriate value element in APL64 command line arguments.

- Command line arguments with argname ‘APL32’, if present, will be used by APL64 to override the settings in the APLNow32 ini-format settings file, if present.
- The setting specified by an APL32 command line argument applies only to the APLNow32 component of APL64, for example the ⎕wi interface.
- Format of the argvalue portion of an APL32 command line argument:

  [Section]Key=Value
- The section corresponds to the analogous section in the APLNow32.ini file of APL64. The section must be enclosed in square brackets.
- The Section, Key and Value elements of an APL32= command line argument cannot contain spaces.
- This command line argument is not saved in the APL64 xml-format configuration file, so it does not apply to future APL64 instances unless it is present as an APL64 command line argument.
- It is possible to have multiple APL32 command line arguments.
- This command line argument applies only to the Windows operating system environment because it pertains to the Win32 components of APL64.
- While an instance of APL64 is running, a temporary APLNow32 xml-format configuration file is created combining the user-provided APL32 ini-format settings file content, if any, and APL64 command line arguments using the format: APL32=[Section]Key=Value.
- Examples:
  - apl32=[MySection1]MyKey1=Value1
  - apl32=[“Section 99”]Key99=Value99
  - apl32=[Config]CTL3D=1
  - a32=[Config]CTL3D=1
  - apl32=[User]SerialNo=123456789
  - apl32=[Config]Wssize=20M

ActivationLicenseId

- Supported argvalue: Non-negative integer
- Default value: 0
- This command line argument is saved in the APL64 xml-format configuration file
- This command line argument applies only to the APL64 developer version
- Upon initialization of an instance of the APL64 developer version:
  - If the software has been previously activated on the workstation, this command line argument is not considered.
  - If the software has not been previously activated on the workstation, this value will be used to attempt on-line activation the software in ‘silent’ mode. If Internet access to the license activation server is not available or if the ActivationLicenseId command line argument is not present or if the provided value is not valid, the activation dialogue will be presented.
- Examples:

    - ActivationLicenseId=123456
    - ALI=123456

ActivationLicensePassword

- Supported argvalue: Text
- Default value: Empty
- This command line argument is not saved in the APL64 xml-format configuration file
- The value of this command line argument is case sensitive
- This command line argument applies only to the APL64 developer version
- This command line argument is used only for the purpose of ‘silent’ activation. It is not used as a default License Password when the activation dialogue is presented.
- Upon initialization of an instance of the APL64 developer version:
    - If the software has been previously activated on the workstation, this command line argument is not considered.
    - If the software has not been previously activated on the workstation, this value will be used to attempt on-line activation of the software. If Internet access to the license activation server is not available or if the ActivationLicensePassword command line argument is not present or if the provided value is not valid, the activation dialogue will be presented.
    - The value of this command line argument is never displayed in the activation dialogs.
- Examples:
    - ActivationLicensePassword=aPL2000!
    - ALP=aPL2000!

ActivationProxyServerUserName

- Supported argvalue: Text
- Default value: ''
- This command line argument applies only if the ActivationWebProxy type is ‘UserEntered’
- This command line argument is saved in the APL64 xml-format configuration file
- The value of this command line argument may be case sensitive
- This command line argument applies only to the APL64 developer version
- This command line argument is used only for the purpose of ‘silent’ activation
- This value will be used to set the User Name field in the Proxy Server section of the Activation dialogs for license validity, refresh the license, activate on-line, or de-activate on-line
- Examples:
    - ActivationProxyServerUserName="John Doe"
    - apsun="John Doe"

ActivationProxyServerUserPassword

- Supported argvalue: Text
- Default value: ''
- This command line argument applies only if the ActivationWebProxy type is ‘UserEntered’

- This command line argument is not saved in the APL64 xml-format configuration file
- The value of this command line argument may be case sensitive
- This command line argument applies only to the APL64 developer version
- This command line argument is used only for the purpose of 'silent' activation. It is not used as a default Proxy Server User Password when the activation dialogue is presented. The value of this command line argument is never displayed in the activation dialogs.
- This value will be used to set the User Password field in the Proxy Server section of the Activation dialogs for license validity, refresh the license, activate on-line, or de-activate on-line
- Examples:
  - ActivationProxyServerUserPwd=abcd
  - APSUP=abcd

ActivationProxyServerUrl

- Supported argvalue: url format text
- Default value: ''
- This command line argument applies only if the ActivationWebProxy type is 'UserEntered'
- This command line argument is saved in the APL64 xml-format configuration file
- The value of this command line argument is not case sensitive
- This command line argument applies only to the APL64 developer version
- This command line argument is used only for the purpose of 'silent' activation
- This value will be used to set the Proxy Server field in the Proxy Server section of the Activation dialogs for license validity, refresh the license, activate on-line, or de-activate on-line
- Examples:
  - ActivationProxyServerUrl=https://mysite.com/
  - Apsu=https://mysite.com/

ActivationWebProxyType

- Supported argvalue: Text
- Default value: None
- Supported values: None, Default, UserEntered
- This command line argument is saved in the APL64 xml-format configuration file
- The value of this command line argument is not case sensitive
- This command line argument applies only to the APL64 developer version
- This command line argument is used only for the purpose of 'silent' activation
- This value will be used to select the Activation Web Proxy Type used to access the activation server to check for license validity, refresh the license, activate on-line, or de-activate on-line.
- Examples:
  - ActivationWebProxyType=None
  - awpt=Default
  - AWPT=userEntered

CheckComponentFile

- Supported argvalue: True or False
- Default value: True
- Argvalue is not case sensitive
- This information is saved in the APL64 xml-format configuration file.
- When argvalue is true:
  - When attempting to tie a file using an APL64 function which is designed to tie an APL64 component file, the target file will be checked to determine if the target file is an APL64 component file.
  - File data integrity checks are suspended.
- This option is provided to assist an APL64 programmer salvage potentially-corrupted component files.
- Examples:
  - CheckComponentFile=0
  - CCF=0

CompFileCommitSync

- Supported argvalue: 0, 1, 2
- Default value: 2
- This information is saved in the APL64 xml-format configuration file.
- This command line argument controls the timing of a component file commit, append or replace, to the physical drive.
- Value 0: Commits are buffered until the next interpreter 'ready' state
- Value 1 or 2: Commits are immediate
- Deferring commits (CFCS=0) may improve system performance, but this setting should be carefully considered when component files are shared by multiple APL64 instances.
- Examples:
  - CompFileCommitSync=0
  - CFCS=1

NavigateDirs

- Supported argvalue: 0, 1
- Default value: 0
- This information is saved in the APL64 xml-format configuration file.
- This command line argument determines if the current directory of the APL64 Developer version will change when specified file actions are performed via menu items, short cuts or system functions/commands:
  - File | Load Workspace
  - File | XLoad Workspace
  - File | Copy Workspace
  - File | PCopy Workspace
  - File | SaveAs
  - Session | Configuration Settings | Export Settings

    - Options | History Log | Import History Log
- This command line argument does not affect file options where the current directory is not modified:
    - Session | Configuration Settings | Import Settings
    - Options | Create Runtime .Net Assembly
    - Options | APL+Win Configuration Conversion
    - Help | License & Activation | APL64 License Activations/Deactivations
- Value 0: The current directory is not modified
- Value 1: The current directory may be modified
- Examples:
    - NavigateDirs=0
    - NAVD=1

EmitLongDateFormat

- Supported argvalue: 0, 1
- Default value: 0
- This information is saved in the APL64 xml-format configuration file.
- This command line argument determines if the APL64 interpreter will emit timestamps in long or short format associated with loading, saving, copying, dropping workspaces and session log.
- Value 0: The short date format will be emitted, e.g. …Friday, March 4, 2022 4:00:29 PM
- Value 1: The long date format will be emitted, e.g. … 3/4/2022 4:00:29 PM
- Examples:
    - EmitLongDateFormat=0
    - ELDF=1

PathCase

- Supported argvalue: 0, 1
- Default value: 0
- This information is saved in the APL64 xml-format configuration file.
- This command line argument determines initial value of the ⎕pathcase system variable when an instance of APL64 is created.
- Value 0: File paths will not be considered case-sensitive by the APL64 interpreter
- Value 1: File paths will be considered case-sensitive by the APL64 interpreter
- Examples:
    - pathcase=0
    - PC=1

Editing of APL64 or APLNow32 Configuration Files

Most APL64 configuration settings in the xml-format configuration file are editable using menu items available in an instance of the APL64 developer version.

The location of the APL64 xml-format configuration file in use during an APL64 developer instance may be obtained using the Session | Settings File Path menu item or the value of the ⎕userpath system variable.

These settings may be exported and imported for use with other APL64 instances on other workstations using the Session | Export Settings and Session | Import Settings menu items. If manual editing of the APL64 xml-format configuration file is anticipated, the use of Notepad++ using the XML plug-in is suggested.

The APLNow32 configuration file uses the ini-file format which is the same as that used by APL+Win.