



FilePath & Placeholders

CODESYS Control – Tutorial

Content

1 Introduction	3
2 Use Cases	3
2.1 How to redirect all files to a dedicated folder or drive	3
2.2 How to redirect application related files	3
2.3 How to use a combination of "FilePath" and "PlaceholderFilePath"	4
2.4 How to include a removable USB flash drive or memory card into the filetransfer dialog	4
2.5 How to access a file via placeholder from IEC	5
2.6 How to redirect the (web) visu files to a larger storage device	5
2.7 How to redirect the trend data base files to a large and fast storage device	5
2.8 How to redirect the alarms data base files to a secure and reliable storage device	5
2.9 How to hide 3S.dat	6
2.10 How to access files that are redirected via wildcards in the filetransfer dialog	6
3 FAQ	6
3.1 How does the runtime sytem resolve these new double placeholders?	6
3.2 What requirements are substantial for Backup&Restore?	7
3.3 How can I reverse the implicit setting "IecFilePath=\$PlcLogic\$" without abandoning the new folder structure by setting "PlcLogicPrefix=-1"?	7
3.4 I forgot the placeholder markers "\$" in my configuration, I configured "PlaceholderFilePath=hi, there", what gives?	7
4 Technical Reference	7
4.1 FilePath	7
4.1.1 Configuration	8
4.2 Placeholders	9
4.2.1 Display of placeholders	9
4.2.2 Configuration	10
4.2.3 Placeholder options	10
4.2.4 Known placeholders	11
4.2.5 New folder structure via PlcLogicPrefix	12
4.2.6 Implicit configuration	12
5 Backup&Restore	13

1 Introduction

FilePath and Placeholders are introduced with the runtime system of CODESYS V3.5 SP8. The general intention of this feature is to group and redirect files on the PLC and completely separate them from runtime system kernel files, i.e., files and folders that are created and accessed via SysFile/SysDir or CAA File interface functions.

With the **"FilePath"** setting the standard path for file access of the runtime system may be changed from its default, the execution directory. Individual files without any paths can be redirected to other paths or folders with their full filename or a group of files with wildcards.

Placeholders are virtual folders to group files or to access files outside of the standard path. In addition this feature is a requirement for the new **Backup&Restore** feature.

In the following chapter special **Use Cases** are discussed. Frequently asked questions are answered in the **FAQ** chapter. For further detail a **Technical Reference** chapter is provided.

In the standard runtime system the configuration of filepaths and placeholders is done in the file "CODESYSControl.cfg" in the section [SysFile]. An empty section means that there are no filepaths but placeholders are implicitly configured, see **Implicit configuration**.

All configured paths are checked at startup of the runtime system and created if necessary, except for placeholders which are configured as **".Volatile"** or **".CreateOnWrite"**.

Important:

You have to keep in mind that paths are handled case sensitive on all platforms. It is highly recommended to use a slash (not a backslash) as folder separator on all systems.

2 Use Cases

2.1 How to redirect all files to a dedicated folder or drive

Configure the following setting to change the standard path:

```
[SysFile]
FilePath=/myfiles
```

In the example all files (files that are accessed via SysFile or CAA File interface functions) are redirected to "/myfiles". Visu files are implicitly redirected into a subdirectory of the standard path when using the old folder structure. In this case visu files are redirected to "/myfiles/visu".

2.2 How to redirect application related files

Using the old flat folder structure, configure:

```
[SysFile]
FilePath.1=/myplc/myapp, *.app, *.crc, *.ret
```

With the old behavior wildcards for application related files still work with V3.5 SP8 and later. The files in the example are explicitly redirected to "/myplc/myapp".

With the **new folder structure** wildcards for application related files are obsolete, because these files are prefixed with the **PlcLogicPrefix**, i.e., "\$PlcLogic\$/<application name>". With the following configuration

```
[SysFile]
PlcLogicPrefix=1
PlaceholderFilePath=/myplc, $PlcLogic$"
```

application related files are redirected to "/myplc/Application", assuming that the application is named "Application".

Note:

With this configuration the whole new folder structure under “\$PlcLogic\$” is redirected to “/myplc”. Single subfolders like “\$visu\$” may be redirected to a different folder additionally, see [New folder structure via PlcLogicPrefix](#).

2.3 How to use a combination of “FilePath” and “PlaceholderFilePath”

Let's start with the redirection of the standard path to “/myfiles” and an explicit placeholder for visu files:

```
[SysFile]
FilePath=/myfiles
PlaceholderFilePath=visu, $visu$
```

In this case we have an absolute standard path for all files and additionally a relative path (i.e., a subdirectory) “visu” for the visu files (which is the default), which resolves to “/myfiles/visu” because of the “FilePath” setting.

Changing the visu configuration to an absolute path looks like this:

```
[SysFile]
FilePath=/myfiles
PlaceholderFilePath=/myvisu, $visu$
```

In this case all visu files are redirected to “/myvisu” regardless of the absolute standard path for all other files.

Using the new folder structure with the same configuration:

```
[SysFile]
PlcLogicPrefix=1
FilePath=/myfiles
PlaceholderFilePath=/myvisu, $visu$
```

In this case all visu files are redirected to “/myvisu” again regardless of the absolute standard path. But where all the other files go depends on their source:

- trend and alarm files will go to “/myfiles/PlcLogic/trend” and “/myfiles/PlcLogic/alarms”
- files from IEC will go to “/myfiles/PlcLogic”
- application related files will go to “/myfiles/PlcLogic/Application”, assuming that the application is named Application.

On top of that using wildcards:

```
[SysFile]
PlcLogicPrefix=1
FilePath=/myfiles
PlaceholderFilePath=/myvisu, $visu$
FilePath.1=texts, *.txt
```

The same as above except for files that have no path and match with the “*.txt” wildcard, they are redirected to “/myfiles/texts” because of the “FilePath” setting.

Conclusion: With “FilePath” the whole file tree can be redirected to a dedicated folder. With “PlaceholderFilePath” one can redirect group of files, and relative filepaths or placeholders are prefixed with the “FilePath”.

For more information see [FilePath](#), [Placeholders](#), [PlcLogicPrefix](#), [New folder structure via PlcLogicPrefix](#).

2.4 How to include a removable USB flash drive or memory card into the filetransfer dialog

Configure

```
[SysFile]
PlaceholderFilePath.1=/mnt/USBDevice, $USBDev$
PlaceholderFilePath.1.Volatile=1
PlaceholderFilePath.1.View=1
```

Since V3.5 SP6 the runtime system checks if this configured path exists. If it does it is displayed as “\$USBDev\$” placeholder in the filetransfer dialog. See placeholder option “**Volatile**” for details.

2.5 How to access a file via placeholder from IEC

Here is an example using the SysFile library. In IEC the ‘\$’ is an escape character. To use the ‘\$’ itself it has to be doubled:

```
VAR
    Result : RTS_IEC_RESULT;
    hFile : RTS_IEC_HANDLE;
END_VAR

hFile := SysFileOpen('$visu$/test.bmp', ACCESS_MODE.AM_READ, ADR(Result));
IF (hFile <> RTS_INVALID_HANDLE) THEN
    SysFileClose(hFile);
END_IF
```

See [Placeholders](#) for details.

2.6 How to redirect the (web) visu files to a larger storage device

Since V3.5 SP6 a “\$visu\$” placeholder is provided to achieve this:

```
[SysFile]
PlaceholderFilePath.1=/mnt/larger_device/, $visu$
```

but this setting is fixed, there is no possibility to change this setting on the fly or in dependency of the presence of a removable device.

2.7 How to redirect the trend data base files to a large and fast storage device

A trend data base can grow very fast to a big file. To regard this one may configure:

```
[SysFile]
PlaceholderFilePath.1=/mnt/fast_device/, $trend$
```

Since V3.5 SP8 the trend data base uses the “\$trend\$” placeholder to access all its files. See [Known placeholders](#) for other predefined placeholders.

2.8 How to redirect the alarms data base files to a secure and reliable storage device

Configure

```
[SysFile]
PlaceholderFilePath.1=/mnt/secure_device/, $alarms$
```

Since V3.5 SP8 the alarms data base uses the “\$alarms\$” placeholder to access all its files. See [Known placeholders](#) for other predefined

placeholders.

2.9 How to hide 3S.dat

Configure

```
[SysFile]
FilePath.1=/hidden_path/, 3S.dat
```

Since V3.5 SP8 an extra "IecFilePath.1=/hidden_path/, 3S.dat" to correctly access "3S.dat" from IEC is not necessary any more, because "3S.dat" is now only controlled by "FilePath". A configuration via "IecFilePath" for "3S.dat" will be ignored. See ["FilePath"](#) for more information.

2.10 How to access files that are redirected via wildcards in the filetransfer dialog

There may be the following configuration:

```
[SysFile]
FilePath.1=/Temp/Wildcards, *.txw
```

To access these files via filetransfer and include them in the backup configure the following placeholder:

```
PlaceholderFilePath.1.View=1
PlaceholderFilePath.1.VParent=$PlcLogic$
PlaceholderFilePath.1=/Temp/Wildcards, $Wildcards$
```

With the option **".View=1"** the placeholder shows up in the filetransfer dialog. With the option **".VParent=\$PlcLogic\$"** the placeholder shows up additionally as child of the **"\$PlcLogic\$"** placeholder.

Every file that is accessible via filetransfer as a child of **"\$PlcLogic\$"** can be backed up with the already implemented **simple backup mode**. On the other hand, any file that is not accessible via filetransfer can not be backed up.

3 FAQ

3.1 How does the runtime sytem resolve these new double placeholders?

Since V3.5 SP8 the visu files for example are accessed with a double placeholder like **"\$PlcLogic\$/visu\$/my.svg"**. The runtime system primarily resolves the first placeholder **"\$PlcLogic\$"**. With this placeholder the whole PlcLogic tree can be redirected to another place than the default: a subdirectory of the execution folder. Next the second placeholder is resolved and it replaces the former one, i.e., the last placeholder wins. In the default configuration of the new folder structure the placeholders are resolved to corresponding subdirectories, i.e., **"\$PlcLogic\$/visu\$/my.svg"** is finally resolved to **"PlcLogic/visu/my.svg"**.

With this double-stage resolving of the **"\$PlcLogic\$/visu\$"** placeholder there is the possibility to redirect just the visu files with the **"\$visu\$"** placeholder or the whole PlcLogic tree with the **"\$PlcLogic\$"** placeholder.

Given the following configuration:

```
[SysFile]
PlcLogicPrefix=1
PlaceholderFilePath=/myplc, $PlcLogic$
PlaceholderFilePath.1=/myvisu, $visu$
```

In this case we have the new folder structure and the first placeholder **"\$PlcLogic\$"** is resolved to **"/myplc"** but the **"\$visu\$"** placeholder is resolved to **"/myvisu"** and this replaces the former resolve. Therefore **"\$PlcLogic\$/visu\$/my.svg"** finally resolves to **"/myvisu/my.svg"**.

Regarding now the compatibility with the old flat folder structure with the double placeholder “\$PlcLogic\$/my.svg”: Primarily the first placeholder “\$PlcLogic\$” is resolved. But because the default configuration of the “\$visu\$” placeholder for the old flat folder structure is just “visu” the double placeholder is finally resolved to “visu/my.svg”.

3.2 What requirements are substantial for Backup&Restore?

- First of all CODESYS and RTS must be at least of V3.5 SP8.
- Then “PlcLogicPrefix=1” must be configured in the [SysFile] section (see [“PlcLogicPrefix”](#)).
- To ensure that files that are redirected via wildcards are included in the backup, see [How to access files that are redirected via wildcards](#).
- The device description must be updated at least to V3.5.80.
- Then there should be a new tab “Backup and Restore” in the “Device” window.
- To “Read backup information from device” there should be at least one application on the device.
- It is recommended to set the device to STOP especially to backup data base files in a consistent state.

See chapter [Backup&Restore](#) for more information.

3.3 How can I reverse the implicit setting “IecFilePath=\$PlcLogic\$” without abandoning the new folder structure by setting “PlcLogicPrefix=-1”?

First of all a note that this undermines the Backup&Restore feature for IEC files with that setting!

With configuring “IecFilePath=” IecFilePath is set to the standard path (“IecFilePath=,” works for this prior to CDS-46893, or V3.5 SP9 respectively).

3.4 I forgot the placeholder markers “\$” in my configuration, I configured “PlaceholderFilePath=hi, there”, what gives?

A folder “hi” will be created. A virtual folder “there” will show up in the “Device/Files” dialog with a normal yellowish folder icon but this virtual folder is not resolved to “hi” because it is not recognized as a placeholder and therefore points into the void. Going into that folder there is not even a “..” entry to go back. By copying a file into “there” an appropriate folder is created in the standard path. Then there are two entries named “there” in the “Device/Files” dialog: the physical one and the virtual one and both they lead to the physical one now. This is probably not the intention.

4 Technical Reference

4.1 FilePath

With the **“FilePath”** setting it is possible to configure the standard path of the runtime system (as known as the root directory, not to be mistaken with the root directory of the underlying file system). Individual files without any paths can be redirected to other paths or folders with their full filename or a group of files with wildcards to meet one’s needs. This configuration can be done differently for C (“FilePath”) and IEC (“IecFilePath”) file access.

These settings concern files and folders that are created and accessed via SysFile/SysDir or CAA File interface functions, except the files are accessed via an absolute path.

If only “FilePath” is configured, file access from IEC falls back to the configuration of “FilePath”. On the other hand if “IecFilePath” is configured only, the file access from C is not affected. When “IecFilePath” is set, there is no fallback to “Filepath” settings. In this case file access from IEC to specific files configured with FilePath is possible only when configured with IecFilePath too. e.g.:

```
[SysFile]
FilePath=/ata0a/
IecFilePath=/ata0b/
FilePath.1=/ata0, test.txt
IecFilePath.1=/ata0, test.txt
```

The last line is necessary when access from IEC is desired. An exception to this is the license file “3S.dat”. Since V3.5 SP8 or CDS-35792

respectively, this file is resolved according to “FilePath” consistent from C as well as from IEC.

Important:

Since V3.5 SP8 and with PlcLogicPrefix=1 (see “PlcLogicPrefix”) “IecFilePath” is implicitly set to “\$PlcLogic\$”!

With the “Force” flags file access can be forced to stay inside the configured paths. All configured paths form a whitelist of allowed paths, i.e., absolute paths that are not in the whitelist and directory traversal paths (beginning with “../”) are prohibited then.

4.1.1 Configuration

“FilePath”

type=String

Setting to configure the standard path for file access, for all filenames with no specified path or for directory access.

The setting has the following structure:

```
FilePath=<directory path, relative or absolute>{[, filedesc]}
```

A filedesc is either a filename or a wildcard to match a specific file extension, it is optional and may be repeated.

Several file paths could be configured, if they were indexed, e.g.:

```
[SysFile]
FilePath.1=./Boot, *.app, *.ap_, *.ret, *.frc
FilePath.2=/Temp/Wildcards, *.txw
FilePath.3=/Temp/Wildcards, File7.txt
```

In this case, files with special extensions or a special name could be separated into several directories. File extensions are specified with the wildcards “*”.

Note:

If the new folder structure is used (see [New folder structure via PlcLogicPrefix](#)) these application related files of “FilePath.1” are now redirected to an application folder (prefixed with “\$PlcLogic/<application>”), i.e., especially these wildcards are obsolete and won’t work any more.

Note:

A double extension like *.trace.csv hasn’t matched at no time.

If only a path but no file wildcard or filename is specified, the standard path for all file or directory access is configured, e.g.:

```
FilePath=/Temp/Default
```

“ForceFilePath”

type=Int

Setting to force the configured file path to every file access.

If an absolute or directory traversal file path is requested and this path is not a configured path an error is returned at this operation (ERR_OPERATION_DENIED).

“IecFilePath”

type=String

Setting to configure the standard IEC path for file access, for all filenames with no specified path or for directory access.

Note:

THIS IS USED ONLY FOR ALL FILE ACCESS ROUTINES OUT OF THE IEC PROGRAM!

Several file paths could be configured, if they were indexed, e.g.:

```
[SysFile]
IecFilePath.1=./Boot, *.app, *.ap_, *.ret, *.frc
IecFilePath.2=/Temp/Wildcards, *.txw
IecFilePath.3=/Temp/Wildcards, File7.txt
```

In this case, files with special extensions or a special name could be separated into several directories. File extensions can be specified with the wildcards “*”.

Note:

If the new folder structure is used (see [New folder structure via PlcLogicPrefix](#)) these application related files of “IecFilePath.1” are now redirected to an application folder (prefixed with “\$PlcLogic/<application>”), i.e., especially these wildcards are obsolete and won’t work any more.

If only a path but no file wildcard or filename is specified, the standard path for all IEC file or directory access is configured, e.g.:

```
IecFilePath=/Temp/Default/IEC
```

Note:

If the new folder structure is used “IecFilePath” is implicitly set to “\$PlcLogic\$” if not configured.

“ForceIecFilePath”

type=Int

Setting to force the configured file path to every file access from IEC.

Note:

THIS IS USED ONLY FOR ALL FILE ACCESS ROUTINES OUT OF THE IEC PROGRAM!

If an absolute or directory traversal file path is requested and this path is not a configured path an error is returned at this operation (ERR_OPERATION_DENIED).

4.2 Placeholders

Placeholders are virtual folders to group files or to access files outside of the standard path (even if “ForceFilePath” is set). They are marked like “\$Placeholder\$”, i.e., the name is enclosed in “\$”. This is mandatory to let the runtime system recognize the placeholder.

Important:

As a placeholder is used in a file path, its name must not contain any characters that are invalid for filenames such as slashes, backslashes, colons, asterisks, question marks and the like.

Placeholders are resolved to their configured paths consistent from C and IEC. If a placeholder can’t be resolved it is replaced by a folder with the placeholder’s name. E.g. “\$Placeholder\$/filename.ext” is replaced with “Placeholder/filename.ext” if “\$Placeholder\$” is not configured.

In IEC the ‘\$’ is an escape character. To use the ‘\$’ itself it has to be doubled:

```
hFile := SysFileOpen('$visu$/test.bmp', ACCESS_MODE.AM_READ, ADR(Result));
```

4.2.1 Display of placeholders

In the “Device/Files” dialog (as known as filetransfer dialog) placeholders have an icon different to the normal yellowish folder icon (assuming the “[View](#)” option is set). Placeholders which lead to subdirectories inside the standard path are internally marked as redundant and their icon is a gray one. This regards to all the implicitly configured placeholders (see [Known placeholders](#), [Implicit configuration](#)). On

the other hand placeholders which are explicitly configured and lead to a different path or even to another device have a blue icon.

All placeholders show up within the standard path, and if a virtual parent is configured they can be found inside their parent placeholder folder too - but not inside the corresponding physical folder. This regards to the implicitly configured placeholders for application related files such as "\$visu\$", "\$trend\$" and the like (see [Known placeholders](#)).

In the "Device/Files" dialog placeholders can't be created, renamed or deleted. These options are disabled.

Independent of the **".View"** option while opening a placeholder folder explicitly the placeholders inside are visible. This is important for the backup of the "\$PlcLogic\$" folder.

In case the **".View"** option is not set, calling SysDirOpen("") and then SysDirRead() successively from IEC one doesn't get the placeholders even one is redirected to "\$PlcLogic\$" because the opening path is not a placeholder. To view the placeholders inside "\$PlcLogic\$" one have to call explicitly SysDirOpen("\$PlcLogic\$").

4.2.2 Configuration

"PlaceholderFilePath"

type=String

Setting to configure a placeholder file path for file access. A placeholder is used like a relative path, but with special placeholder marks, e.g.: "\$visu\$/mainwindows.bmp"

The setting has the following structure:

```
PlaceholderFilePath=<directory path, relative or absolute>, <placeholder>
```

Example:

```
[SysFile]
PlaceholderFilePath=/MyPlace/, $MyPlace$
```

Several file paths could be configured, if they were indexed, e.g.:

```
PlaceholderFilePath.1=visu/, $visu$
```

Note:

The trailing slash of the path is optional, it is added automatically.

4.2.3 Placeholder options

".View"

type=Int

Setting to make a single placeholder visible in the file transfer window. The default setting is 0 (invisible).

Example:

```
[SysFile]
PlaceholderFilePath=/MyPlace/, $MyPlace$
PlaceholderFilePath.View=1
```

"PlaceholderFilePath.All.View"

type=Int

Setting to make all placeholders visible in the file transfer window at once. The default setting is 0 (invisible).

A single placeholder can be hidden by setting its **".View"** option to 0.

Example:

```
[SysFile]
PlaceholderFilePath=/MyPlace/, $MyPlace$
PlaceholderFilePath.1=visu/, $visu$
```

```
PlaceholderFilePath.2=/Temp, $Temp$
PlaceholderFilePath.All.View=1
PlaceholderFilePath.2.View=0
```

In this example the placeholders “\$MyPlace\$” and “\$visu\$” are visible in the file transfer window while “\$Temp\$” is not. But all three placeholders are accessible e.g. from IEC.

“.Volatile”

type=Int

Setting to mark a placeholder as volatile, i.e., the specified folder is not created automatically at startup and the placeholder is visible in the filetransfer windows only when the folder exists. The default setting is 0 (not volatile).

This behavior is important for hot plug devices like USB flash drives. Therefore the configured path is not created on write access.

Example:

```
[SysFile]
PlaceholderFilePath.2=/mnt/USBDevice, $USBDev$
PlaceholderFilePath.2.Volatile=1
PlaceholderFilePath.2.View=1
```

Note:

The “.View” option is not implicitly set.

“.VParent”

type=String

Setting to virtually connect a placeholder with a parent placeholder, i.e., in the filetransfer window this placeholder is reachable via its virtual parent placeholder additionally.

Example:

```
[SysFile]
PlaceholderFilePath.1=/temp/oem, $oem$
PlaceholderFilePath.1.VParent=$PlcLogic$
PlaceholderFilePath.1.View=1
```

The oem placeholder is virtually placed under “\$PlcLogic\$” but physically at /temp/oem.

Note:

The “.View” option is not implicitly set.

“.CreateOnWrite”

type=Int

Setting to mark a placeholder so that the specified folder is created when a file is to be written in that folder, the folder is not created at startup, the placeholder is visible in the filetransfer windows only when the folder exists. The default setting is 0 (create at startup).

Example:

```
[SysFile]
PlaceholderFilePath.2=/Temp, $Temp$
PlaceholderFilePath.2.CreateOnWrite=1
PlaceholderFilePath.2.View=1
```

Note:

The “.View” option is not implicitly set.

4.2.4 Known placeholders

“\$PlcLogic\$”

"\$visu\$"
 "\$trend\$"
 "\$alarms\$"
 "\$ac_persistence\$"
 "\$TBF\$"
 "\$_cnc\$"

All these placeholders are configured implicitly at startup. And then they are created except "\$TBF\$" which has the ".CreateOnWrite" option set. Their path depends on the "**PlcLogicPrefix**" setting.

4.2.5 New folder structure via PlcLogicPrefix

"PlcLogicPrefix"

type=Int

Setting to configure the plc logic placeholder prefix that is used now by all application related files.

-1 - old behaviour: no prefix and a flat file structure (default):

```

[PlcLogic$]
[visu$]
[_cnc$]
all other known placeholders are set to the standard path
  
```

0 - migration: starting with old behaviour (no prefix and a flat file structure), but "Reset Origin Device" or a startup with no bootproject will install the new folder structure and change that value to 1

1 - new folder structure:

```

[PlcLogic$]
+ [<application>]
+ [trend$]
+ [alarms$]
+ [ac_persistence$]
+ [visu$]
+ [_cnc$]
  
```

With the new folder structure ...

- file access from IEC is redirected to "\$PlcLogic\$". To achieve this "IecFilePath=\$PlcLogic\$" is implicitly set, if "IecFilePath" is not explicitly configured otherwise.
- all application related files are prefixed with "\$PlcLogic\$/<application>", i.e., a redirection with wildcards (see "**FilePath**") does not work for these files any more. Application related files are:

```

- <application>.app
- <application>.crc
- <application>.ret
- <application>.frc
- <application>.xml
- <application>.core
- <application>.trace.csv
  
```

- the new feature **Backup&Restore** is enabled.

4.2.6 Implicit configuration

If not configured otherwise the implicit file path configuration is as follows:

```

[SysFile]
#if (PlcLogicPrefix==1)
IecFilePath=$PlcLogic$
PlaceholderFilePath=PlcLogic/, $PlcLogic$
PlaceholderFilePath.1=$PlcLogic$/visu/, $visu$
  
```

```

PlaceholderFilePath.1.VParent=$PlcLogic$
PlaceholderFilePath.2=$PlcLogic$/trend/, $trend$
PlaceholderFilePath.2.VParent=$PlcLogic$
PlaceholderFilePath.3=$PlcLogic$/alarms/, $alarms$
PlaceholderFilePath.3.VParent=$PlcLogic$
PlaceholderFilePath.4=$PlcLogic$/ac_persistence/, $ac_persistence$
PlaceholderFilePath.4.VParent=$PlcLogic$
PlaceholderFilePath.5=TBF/, $TBF$
PlaceholderFilePath.5.CreateOnWrite=1
PlaceholderFilePath.6=$PlcLogic$/_cnc/, $_cnc$
PlaceholderFilePath.6.VParent=$PlcLogic$
#else
IecFilePath=
PlaceholderFilePath=PlcLogic/, $PlcLogic$
PlaceholderFilePath.1=visu/, $visu$
PlaceholderFilePath.2=, $trend$
PlaceholderFilePath.3=, $alarms$
PlaceholderFilePath.4=, $ac_persistence$
PlaceholderFilePath.5=TBF/, $TBF$
PlaceholderFilePath.5.CreateOnWrite=1
PlaceholderFilePath.6=_cnc/, $_cnc$
#endif

```

Individual known placeholders may be explicitly configured which override the single implicit setting such as:

```

[SysFile]
PlaceholderFilePath=/mnt/slow_device/, $PlcLogic$
PlaceholderFilePath.1=/mnt/fast_device/, $trend$
PlaceholderFilePath.2=/mnt/secure_device/, $alarms$

```

The above listed implicit ".VParent" setting will be preserved until not otherwise configured. E.g.:

```

PlaceholderFilePath.3=/somewhere/, $_cnc$
PlaceholderFilePath.3.VParent=

```

In this case the placeholder "\$_cnc\$" is no longer found under "\$PlcLogic\$" but in the standard path only.

If "IecFilePath" is explicitly configured, than an extra Placeholder "\$IecFilePath\$" is added in order to ensure access to IEC files via file transfer.

5 Backup&Restore

The new feature of V3.5 SP8 "Backup&Restore of application related files" has the aim to backup and restore all files an application deals with, i.e., the application itself and all files it reads or writes, such as visu files, data base files, the retain variable file, all files accessed via IEC.

To bundle all these files a new folder and file structure is introduced:

```

[$PlcLogic$]
+ [<application>]
- <application>.app
- <application>.crc
- <application>.ret
- <application>.frc
- <application>.xml
- <application>.core

```

```

- <application>.trace.csv
- <files created out of IEC>
- IoConfig.par
+ [$trend$]
+ [$alarms$]
+ [$ac_persistence$]
+ [$visu$]
[$TBF$]

```

To switch to this new folder and file structure “PlcLogicPrefix=1” must be configured in the [SysFile] section (see “PlcLogicPrefix”).

With the simple mode implemented in V3.5 SP8 CODESYS fetches the contents of the “\$PlcLogic\$” folder and all its subfolders (either physical or virtual, except the redundant ones) via file transfer from the RTS. The files are then put into an archive with a .tbf extension. TBF stands for Target Backup File. The restore of all the application related files works vice versa. Backup and restore of the complete runtime system will be provided in the future within CDS-45479.

A special treatment is done to retains which are written to a file by the RTS just before the backup information is given to CODESYS.

Datebase files may be not consistent when the RTS is not in STOP.

The folder “\$TBF\$” is considered to hold backup archives on the RTS file system and CODESYS looks there and displays these archives as a choice to restore from. This folder is not created by the RTS at startup but only when an archive is stored in there.

This chapter is just a short summary of this new feature. A separate documentation will be provided in the future with CDS-44812.

Version History

Version	Description	Author	Date
0.1	Creation	CD	12.01.2016
0.2	Revision and expansion, prepared for presetaion at runtime meeting	CD	25.05.2016
0.3	Revision, admonitions	CD	30.05.2016
0.4	Revision, tips from TZ, replaced H4 and links	CD	03.06.2016
1.0	Review and Release	AH	07.06.2016
1.1	Added legal note	TZ	23.08.2016
2.0	Release	MaH	23.08.2016
2.1	Legal note removed	GeH	24.10.2016
3.0	Release	MN	28.11.2016