



EDS Guide for ACx550/ACS800 and RDNA/RCNA/RETA

Handled by
Drive Care

Date
6.5.2009

Pages
1 (7)

What are EDS files?

EDS stands for Electronic Data Sheet, meaning a file on disk that contains configuration data for a specific device. EDS files are needed when implementing a DeviceNet or ControlNet fieldbus using, for example, Rockwell Software RSNetWorx. The DeviceNet EDS files also describe the parameters of the device, making it possible to configure the device via the fieldbus. However, this is not possible with RSNetWorx for ControlNet or Ethernet/IP.

Selecting the ABB EDS file type

Due to the static nature of DeviceNet, ControlNet and Ethernet/IP EDS files and the vast adaptability of the ABB drives, a single EDS file can't cover all configurations of a drive model. As the parameter groups of the drive are changed by some parameters (i.e. motor ctrl mode), the EDS file needs to be changed as well because some programs aren't compatible with EDS files containing parameters that are invisible on the device.

There are three types of EDS files for ACx550/ACS800 and RDNA/RCNA:

1. Typical – offers the best compatibility with different configurations.
2. Extended – enables configuration of the drive via the fieldbus.
3. Custom – needed when some special parameters need to be changed.

Note! The different types of EDS files have the same product code ID and Vendor code. Therefore you cannot have more than one type of EDS for a single drive model registered concurrently.

Typical EDS files

The typical EDS files contain no information on the drive parameters and therefore are compatible with all drive configurations. On the other hand, the drive has to be configured using the control panel or a separate configuration tool. The typical EDS files are named such as

ACS550-xx_RDNA01_appl1.36_Typical_filerev3.1.EDS

ABB Oy

Postal Address:
Drives
P.O.Box 184
FIN-00381 Helsinki, Finland

Street Address:
Drives
Hiomotie 13
FIN-00380 Helsinki, Finland

Telephone:
+358 10 2211
Telefax:
+358 10 22 xxxxx

Internet:
www.abb.fi
e-mail:
first name.last name
@fi.abb.com

Business Identity Code:
0763403-0
Domicile: Helsinki

Extended EDS files

The extended EDS files contain a definition of the drive parameters that can be modified using RsNetWorx. The files, however, are compatible only with the specific application, communication profile and device configuration. When changing some drive parameters, you may have to unregister the EDS file and register another. The extended EDS files are named such as

ACS550-Ux_RDNA01_appl1.36_Extended_Vector_ABBDrives_filerev3.1.EDS

Note! RSNetWorx for DeviceNet supports device parameters.

Note! The default values of parameters depend on other parameters and the characteristics of the device. It's highly recommended not to use the default values specified in the EDS file and always upload the parameter configuration from the device before making any changes.

EDS file revisions 3.xx and 4.xx

Due to the hardware of the RDNA-01 module being updated, the major revision number has been updated from 1 to 2. Many PLC configuration programs (including RSNetWorx) require that the major revision number of the module and the EDS file must match. This means that new EDS files have been made for RDNA-01 version 2.xx, and these files are not backwards compatible with RDNA-01 versions 1.xx.

For this reason there are now two different EDS file packages, separated by the file revision number. The packages with file revision 3.xx are for use with the older RDNA-01 versions (as well as RCNA-01 and RETA-01), and file revision 4.xx is for use with the newer RDNA-01 versions.

Choose an EDS file with a file revision 3.xx when...

Using RDNA-01 rev. J or earlier

Using RCNA-01

Using RETA-01

Choose an EDS file with a file revision 4.xx when...

Using RDNA-01 rev. K or later

Naming of the EDS files

The ACS550 files are named according to the following scheme:

ACS550-<region>_<fba>_<fba version>_<scope>_<mode>_<profile>_<file revision>.EDS

The ACS800 files are named according to the scheme

ACS800_<application>_<fba>_<fba version>_<scope>_<mode>_<profile>_<file revision>.EDS

Field	Explanation	Examples
Region	Is the ACS550 drive European or US model	0x for European Ux for US
Application	The application of the ACS800 drive.	Standard (drive software ASXR...) Crane (drive software ACXR...) MotionControl (drive software APXR...) etc.
Fba	The type of the fieldbus adapter	RDNA01

		RCNA01 RETA01
Fba version	The firmware version of of the fieldbus adapter. Note! The file can also be used with older firmwares.	appl1.23 (printed on the label behind the module)
Scope	The type of the EDS file	Typical Extended Custom
Mode	Motor control mode of the drive	Scalar VectorTorq (ACS550 only) VectorSpeed (ACS550 only) DTC (ACS800 only)
Profile	The communication profile being used	Generic (for instances 20,21,70 and 71) ABBDives (for other instances)
File revision	Version of the EDS file	filerev1.0 (for example)

Some examples of file names:

Typical EDS file for ACS800 crane application and DeviceNet:

ACS800_Crane_RDNA01_appl1.36_filerev3.3.EDS

Extended EDS file for ACS800 standard application and DeviceNet when using scalar motor control mode and instances 20/70 or 21/71:

ACS800_Standard_RDNA01_appl1.36_Extended_Scalar_Generic_filerev3.3.EDS

Extended EDS file for ACH550 US model and DeviceNet when using vector: torque motor control mode and instances 100/101:

ACH550-Ux_RDNA01_appl1.36_Extended_VectorTorq_ABBDrives_filerev3.1.EDS

Solving problems with extended EDS files**Uploading parameters from drive fails**

There are some occasions when an extended EDS file does not work and uploading parameters from the drive fails with an error like this:

"Address NN: Communication Error(0xC) 'Error response received: [0x040c, 0x00ff]. Object state conflict.', SCIA(0xE, 0x29, 0x1, 0x5), Param10"

There are two possible reasons for this: the drive may be configured incorrectly or the drive may not support the parameter at all.

If you are planning to use the drive with, for example, the ABB communication profile but the drive has not yet been configured to use it, the configuration software may fail when trying to fetch some parameters that aren't enabled in the drive yet. This problem can be solved by changing the appropriate parameters using the drive panel and making sure that the drive configuration conforms to the EDS file name. Check if the fieldbus module is enabled (group 98), the input and output instances (group 51), communication profile (param 98.07, ACS800 only) and motor ctrl mode (group 99). (See section *Naming of the EDS files*)

If you are sure that you are using the correct EDS file and the drive is configured properly, it is possible that the parameter is not needed and not even supported by your drive version. In this case you may have to modify it to meet your drive configuration. It is possible to change the visibility of a parameter in the Parameters window by altering the EDS file. (See below)

Downloading a parameter to drive fails

Downloading a parameter to the drive can fail because of many reasons:

- The drive is configured incorrectly or the drive does not support the parameter. See section *Uploading parameters from drive fails* above.
- The parameter may be write protected at the moment. This occurs with some parameters when the drive is running and usually RSNetWorx shows error message *Attribute not settable*. Try changing the value by the drive panel. If changing fails, the parameter is write protected.
- Some parameters can't be set to their maximum or minimum values via fieldbus due to value rounding and boundary checks.

A parameter is listed in manual but not shown in the parameter window

Most parameters not supported by all drive versions or configurations are defined to be invisible as they might cause problems when uploading or downloading the parameter values. The parameter information is, however, available in the EDS file and it can be enabled when needed. Refer to the next section to enable this kind of parameters.

Hiding unnecessary parameters that cause problems or showing hidden parameters

You can change the visibility of a parameter by opening the EDS file in a text editor such as Microsoft Notepad, and searching for a section that begins with Param keyword and describes the desired parameter. For instance, the section of the Param10 mentioned above looks something like this:

```

$$$$      Net Control
Param10 =
          0,          $ data slot - don't care
          6,          $ path length
          "20 29 24 01 30 05",          $ path (class, instance, attribute)
          0x0002,      $ descriptor: 0x0002=use param, 0x0202=hide param
          0xC1,        $ data type BOOL
          1,          $ data size
          "Net Control",
          "",          $ unit
          "See Firmware Manual, Net Control",
          0,1,0,      $ min, max, default
          ,, ,        $ mult, div, base, offset scaling
          ,, ,        $ mult, div, base, offset links not use
          0;          $ decimal places

Enum10 =
          0, "LOCAL CONTROL", 1, "NETWORK CONTROL";

```

The visibility of the parameter in the Parameters window is determined by bit 9 (0000 0010 0000 0000 or 0x200 in hex) of the descriptor field. If the bit is 0 the parameter is visible in the list and if the bit is 1 the configuration software ignores the parameter. In practice, you can change the visibility by replacing the value of the descriptor field (0x0002) with the hide param value given in the comment (0x0202 in this case). Correspondingly, you can change the value if you want to enable a parameter that is invisible by default.

Note! After editing an EDS file, it is recommended that you save it with a different name than the original file. Replace the word 'Extended' with 'Custom' to make it clear that the file has been edited.

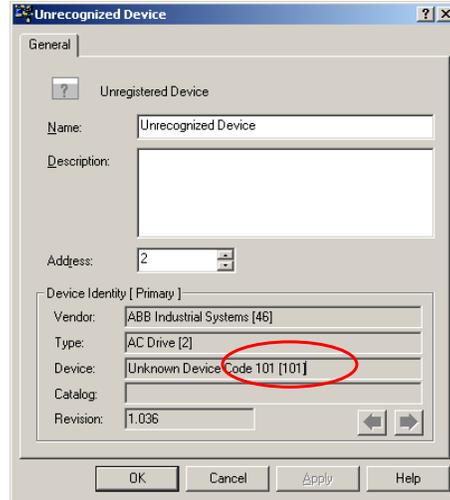
There are no EDS files available for the ACS800 application in use and RSNetWorx shows an Unrecognized Device

As fieldbus support is added in the various ACS800 applications, we pursue creating the typical EDS files for them as soon as possible. Sometimes, however, the EDS files may not be available. Luckily, it is easy to modify the Typical EDS file of the Standard application to suit the special application drive.

First of all, you need to find out the device code of the drive. When you try to open the Properties dialog for an Unknown Device in RSNetWorx, a popup window appears:



If you answer No here, you can see the Properties dialog with the device code.



Now make a copy of the typical EDS file of the standard application and rename it for your application. Then edit the file by opening the EDS file in a text editor such as Microsoft Notepad:

[File]

```
DescText="ACS800 (Standard) with RDNA01";
CreateDate=07-17-2006;
CreateTime=12:00:00;
ModDate=08-01-2006;
ModTime=09:00:00;
Revision=3.1;
```

Type the name of your application here

You can change the modification date and time of the file (optional)

[Device]

```
VendCode=46;
VendName="ABB Automation Inc.";
ProdType=2;
ProdTypeStr="AC Drive";
ProdCode=101;
MajRev=1;
MinRev=36;
ProdName="ACS800 (Standard) with RDNA01";
Catalog="ACS800";
```

Change the device code here

Type the name of your application here

RSNetWorx for ControlNet crashes or hangs

All ControlNet EDS files contain a section called Connection Manager, although it is not used with all ControlNet scanners (Allen Bradley 1756 series, for example). RSNetWorx for ControlNet is known to hang sometimes if the EDS file contains a connection that is not used, so the file may have to be edited.

You can try to solve the problem by opening the EDS file in a text editor such as Microsoft Notepad and removing the following sections (Params and Connection Manager):

```
[Params]
Param1 =
    0,                $ first field shall equal 0
    ,,               $ path size,path
    0x0000,          $ descriptor
    0xC7,            $ data type : 16-bit Unsigned Integer
    2,               $ data size in bytes
    "Output Size",  $ name
    "",             $ units
    "",             $ help string
    4,30,4,         $ min,max,default data values, 0,450,16,
    0,0,0,0,        $ mult,dev,base,offset scaling not used
    0,0,0,0,        $ mult,dev,base,offset link not used
    0;              $ decimal places not used

Param2 =
    ...             section continues ...

Param3 =
    ...             section continues ...

Param4 =
    ...             section continues ...

[Connection Manager]
Connection1 =
    0x04010002,     $ trigger & transport
                   $ 0-15 = supported transport classes
                   $ 16 = cyclic (1 = supported)
    ...             section continues ...

    "",             $ Help string
    "20 04 24 01 2C [Param3] 2C [Param4]"; $ exclusive output path
    1_PLC5C_RTD_Format1 = 1_PLC5C_1794_Discrete_RTD_Format;
```

Note! After editing an EDS file, it is recommended that you save it with a different name than the original file. Replace the word 'Typical' with 'Custom' to make it clear that the file has been edited.