DEFINE WORK FILE

```
DEFINE WORK FILE n  ⎰ operand1 [TYPE operand2] ⎱  [ATTRIBUTES {operand3}...]
                    ⎱ TYPE operand2             ⎰
```

Note:
The elements shown in square brackets [...] are optional, however, at least one of them must be specified with this statement.

This chapter covers the following topics:

- Function

- Syntax Description

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Related Statements: CLOSE WORK FILE | READ WORK FILE | WRITE WORK FILE

Belongs to Function Group: *Control of Work Files / PC Files*

Function

The statement DEFINE WORK FILE is used to assign a file name to a Natural work file number within a Natural application.

This allows you to make or change work file assignments dynamically within a Natural session or overwrite work file assignments made at another level.

When this statement is executed and the specified work file is already open, the statement will implicitly close the work file.

Note:
For Unicode and code page support, see *Work Files and Print Files on Windows, UNIX and OpenVMS Platforms* in the *Unicode and Code Page Support* documentation.

Syntax Description

Operand Definition Table:

Operand	Possible Structure				Possible Formats											Referencing Permitted	Dynamic Definition
operand1	C	S			A	U										yes	no
operand2	C	S			A	U										yes	no
operand3	C	S			A	U										yes	no

Note:

If a format U operand is specified in Unicode (UTF-16), it is converted to session code page characters before it is evaluated.

Syntax Element Description:

Syntax Element	Description
DEFINE WORK FILE *n*	**Work File Number:** *n* is the work file number (1 to 32). This is the number to be used in a WRITE WORK FILE, READ WORK FILE or CLOSE WORK FILE statement.
operand1	**Work File Name:** *operand1* is the name of the work file. The file name (*operand1*) may contain environment variables. It is possible to use physical work file names. If a file with the specified name does not exist, it will be created. If *operand1* is not specified, the value of *operand1* is determined by taking the work file name stored with the Configuration Utility in the parameter file for the corresponding work file number. **Note:** If *operand1* is not specified, the behavior of Natural for Mainframes and Natural for Windows/UNIX/OpenVMS is different.

Syntax Element	Description	
`TYPE operand2`	**TYPE Clause:** `operand2` specifies the type of work file. The value of `operand2` is handled in a case insensitive way and must be enclosed in quotes or provided in an alphanumeric variable.	
	`DEFAULT`	Determines the file type from the extension. Format: Depends on the work file type. **Note:** The file type `TRANSFER` cannot be determined by the work file type `DEFAULT`. You must explicitly define `TRANSFER` as the file type you wish to use.
	`TRANSFER`	Is used to transfer data to and from a PC with Entire Connection. This work file type represents a data connection between a Natural session on UNIX or OpenVMS and an Entire Connection terminal on a PC. Format: `ENTIRE CONNECTION` **Notes:** 1. This work file type cannot be used in conjunction with the *ATTRIBUTES Clause*. 2. This work file type is not available under Windows.
	`SAG`	Format: binary
	`ASCII`	Files in ASCII are "text" files with records terminated by [a carriage return] line feed. Format: ASCII
	`ASCII-COMPRESSED`	Is a file in ASCII format, with the exception that all trailing blanks are removed. Format: ASCII
	`ENTIRECONNECTION`	With this work file type, you can read and write (using the statements `READ` and `WRITE`, for example) directly from/to a work file in Entire Connection format on the local disc. Format: `ENTIRE CONNECTION` **Note:** This work file type is available on PCs, on UNIX and on OpenVMS. No transfer to PC is possible. The Entire Connection terminal is not used in this process.
	`UNFORMATTED`	A completely unformatted file. No formatting information is written (neither for fields nor for records). Format: `UNFORMATTED`
	`PORTABLE`	Files which can handle dynamic variables exactly and can also be transported: for example, from a Little Endian machine to a Big Endian machine, and vice versa. Format: `PORTABLE`
	`CSV`	Comma-separated values. Each record is written to one line in the file. By default, a header is not written. The default character which is used to separate the data fields is a semicolon (;). For further information, see *Work Files* in the *Configuration Utility* documentation.

Syntax Element	Description

ATTRIBUTES
{*operand3*}...
	ATTRIBUTES Clause:
	
	operand3 specifies a work file attribute.
	
	Several attributes separated by a comma or a blank can be specified, for example:
	
	`DEFINE WORK FILE ATTRIBUTES 'APPEND,KEEP'`
	
	If multiple values for the same attribute type are specified, the last value is taken, for example:
	
	`DEFINE WORK FILE ATTRIBUTES 'APPEND,NOAPPEND'`
	
	In this case, NOAPPEND will be taken.
	
	Example for BOM/NOBOM usage:
	
	<pre>...
	DEFINE WORK FILE 11 'x.tmp' ATTRIBUTES 'BOM'
	*
	* write work file with BOM
	*
	DEFINE WORK FILE 11 'x.tmp' ATTRIBUTES 'NOBOM'
	*
	* write work file without BOM
	...</pre>
	Note:
	If *operand3* is omitted, the corresponding value defined in the parameter file, as created by the Configuration Utility, is implicitly used.
	
	The following is an overview of the attribute types and their possible values:

Enable/Disable Append Mode:	
APPEND	Activates the append mode. In this mode, new records are added at the end of the file.
NOAPPEND	Deactivates the append mode. The file is rewritten from the start. This is the default value.

Enable/Disable Write BOM Mode:	
BOM	Activates the write BOM (byte order mark) mode. A BOM will be written that precedes the work file data.
NOBOM	Deactivates the write BOM mode. No BOM will be written. This is the default value.

Keep/Delete File after Work File Close:	
DELETE	The work file is deleted after a close work file operation.
KEEP	The work file is kept after a close work file operation. This is the default value.