

# EXECUTE

## Description

Execute a procedure file.

You can nest up to 7 procedure files.

## Syntax

```
EXECUTE path [procedurefileparameter] ...
```

*path* is as follows:

```
{[[drive:]\directory\...\]filename[.extension]}
```

If you do not specify an *extension*, the extension "ncp" is automatically appended to the procedure file name.

If you do not specify a *drive* and/or *directory*, Entire Connection tries to locate the file in the current drive and directory. If the file cannot be found, Entire Connection then tries to locate the file in the procedure directory which has been defined in the user properties.

You can specify up to 9 *procedurefileparameters* (separated by blanks) that correspond the input parameters of the specified procedure file. These are stored in the local variables #PARM1 through #PARM9. The variable #PARM0 contains the full path of the executed procedure file. The variable #PARMNO contains the number of passed parameters passed (00 to 09).

## Examples

- Execute the procedure file *Test1.ncp* without parameters:

```
EXECUTE Test1
```

- Execute the procedure *Myproc* and using the parameter value contained in the local variable #PARM1:

```
EXECUTE Myproc #PARM1
```

- Execute the procedure file *Test2.abc* with two parameters:

```
EXECUTE Test2.abc one two
```

- Execute the procedure file *\MYDIR\Test3.ncp* with three parameters:

```
EXECUTE \MYDIR\Test3 one two three
```

- Execute the procedure file which is defined by the local variable #PARM1 without parameters:

```
EXECUTE #PARM1
```

## Procedure File Examples

*Ncpnest.ncp, Findfile.ncp*

## Variables Returned

#RC (SUCCESS if there was no runtime error. FAILURE if there was a runtime error.)

## Usage

Procedure File:	Yes
Command Line:	Yes
Key:	Yes
API:	Yes