Emacs support for NetRexx

Description	Keystroke	Function	Note
NetRexx Programming Language Support See also **𝔓 - REXX**	Support for the **NetRexx programming language** is minimal: you can activate the netrexx-mode package by setting the **pel-use-netrexx** user option. • Files with the .rexx, .elx, .ncomm and .cpr are recognized as REXX source files and will automatically activate forth-mode if the package has been activated via that user-option. • Generic programming language features like template text insertion handle NetRexx comment style. See **∑ Inserting Text** . • 📦 NetRexx support is provided by **netrexx-mode** external package 🔄 automatically downloaded and installed by PEL when the **pel-use-netrexx** user option is set to **t**.		
Open this PDF file. See also: **∑ Help/ Info**	**<f11> SPC N <f1>** **<f12> <f1>**	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the local copy of the **𝔓 - NetRexx** PDF file unless a command prefix (like **C-u**) was used. In that case it opens the Github-hosted file instead.
∑ Customize PEL NetRexx support	**<f11> SPC N <f2>** **<f12> <f2>**	(**pel-customize-pel** &optional OTHER-WINDOW)	Customize PEL NetRexx support. Use this to activate NetRexx support. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
∑ Customize Emacs NetRexx support	**<f11> SPC N <f3>** **<f12> <f3>**	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs NetRexx support: netrexx-mode • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Editing NetRexx Code	NetRexx support includes: • font locking • return handling, indentation control • imenu/speedbar support for .nrx files showing class and methods. • Navigation across methods • character pairing ☛ All commands accessible via the **<f12>** mode sensitive key prefix are available via the **<f11> SPC N** global key prefix.		
Indent NetRexx code	**<tab>**	(**netrexx-indent-line**)	Indent the current line as NetRexx code. • For more information, type **<f1> k <tab>** to read the command docstring.
Move point to next method definition	• **C-c C-n** • **<f12> <down>**	(**netrexx-next-method**)	Jumps to the next method definition.
Move point to previous method definition	• **C-c C-p** • **<f12> <up>**	(**netrexx-previous-method**)	Jumps to the previous method definition.
Move to beginning of method	**C-M-a**	(**netrexx-beginning-of-method** &optional ARG)	Jumps to the beginning of the method. • ARG repeats the search ARG times. It always returns t, unless no method is found.
Move to end of method	**C-M-e**	(**netrexx-end-of-method** &optional ARG)	Jumps to the end of the method. • ARG repeats to search ARG times. It always returns t, unless no method end is found. • Comments before the method are reckoned to be part of that method. Meaning that if point is at a comment that describes a method, this function will bring you to the end of that method.
Special pairing keys	• **(** • **{** • **[** • **'** • **"**	(**skeleton-pair-insert-maybe** ARG)	Insert the character you type ARG times. • Without arguments the matched paired character is inserted and the point is left between the two characters. • For **(** the **( )** characters are interred, **{** yields **{ }**, etc… • With a numeric argument that number of characters are inserted, without pairing. • Therefore to insert only one opening parenthesis character type **M-1 (**
Select current block of code	**<f12> =**	(**netrexx-select-current-block**)	Selects all lines between matching do (loop / select) and end. • It will return t if it can find an "end" statement below the point and that "end" statement has a matching "do", "loop" or "select" statement. • If it cannot find such a statement, it will select the whole method. • Belonging to that method are the comments written directly before the method statement. Normally these are the javadoc style comments, but it could be any kind of comment. This means that if point is at a line that contains a comment, it will skip forward until it finds a non-comment line. It will then select the whole method, including the comments before the method statement.
Sanitize code in marked region	**<f12> s**	(**netrexx-sanitize-region** BEG END)	Removes double empty lines and trailing whitespaces and will comment out all "trace results" and "trace methods" statements that are not part of an "if" statement. All other lines are indented with TAB. All blank lines between a multi-line comment and a method are removed.
Insert end e=comment	**<f12> ;**	(**netrexx-insert-end-comment**)	Inserts a comment right after an netrexx "end" statement that shows which "do", "loop" or "select" it matches.
Insert end comment for the marked region	**<f12> e**	(**netrexx-insert-end-comment-region** BEG END)	Every end-statement in the region that matches M-x netrexx-looking-at-end-p will get an end comment. • See also **<f12> ;**.
Insert javadoc comment for the method	**<f12> j**	(**netrexx-insert-javadoc-for-method**)	Inserts an appropriate javadoc statement for the method. • The javadoc based on: - the name of the method, - the name and type of the parameters, - the return type of the method.

NetRexx— References

Document	Notes
NetRexx	A REXX language derivative that integrates ideas from Object Rexx and Java. Free License, open source. Runs on the JVM. See also **𝔓 - REXX**
NetRexx @ Wikipedia	
Emacs support for NetRexx	
netrexx-mode.el @ netrexx.org	A 2003 implementation that supports Net-Rexx on Emacs…
netrexx-mode.el @ GitHub	…and its mirror in GitHub.
prouleau/netrexx-mode.el @ GitHub	… and my updated version, on GutHub, that fixes byte-compiler warnings, used by PEL.