Conventions

This document covers the following topics:

- Syntax Conventions

- Adabas Utility Syntax

Syntax Conventions

The following table describes the conventions used in syntax diagrams of Adabas statements, commands, and ADARUN parameters.

Convention	Description	Example
uppercase, bold	Syntax elements appearing in uppercase and bold font are Adabas keywords. When specified, these keywords must be entered exactly as shown.	`ADADBS  CHANGE  FILE = file-number` The syntax elements ADADBS, CHANGE, and FILE are Adabas keywords.
lowercase, italic, normal font	Syntax elements appearing in lowercase and italic font identify items that you must supply.	`ADADBS  CHANGE  FILE = file-number` The syntax element *file-number* identifies and describes the kind of value you must supply. In this instance, you must supply the number of the file affected by the ADADBS CHANGE operation.

Convention	Description	Example

mixed case, normal font	Mixed case is used for two purposes:
	
	1. To identify the short form of a keyword. The short form is the part in uppercase; the lowercase parts of the statement, command, or parameter is not necessary.
	
	   Note:
	   If a lowercase part appears in italics, it represents a value you must supply; in this case the syntax would not identify a short form.
	
	2. Some values of parameters are taken from values of other parameters. In these cases, the name of the parameter referenced for the value appears in uppercase
	
	Syntax elements appearing in mixed case and normal font (not bold or italic) identify items established by other Adabas control statements. This notation is usually used to identify how default values are determined for some parameters in Adabas syntax.
		1. **DEvice**
		
		   In the example above, the minimum form of the parameter is DE.
		
		   However, in the following example, the *nn* identifies a value you must supply. There is no minimum form of the parameter.
		
		   UEX*nn*
		
		2. [SORTDEV = { *device-type* | ADARUN-device }]
		
		   The syntax element "ADARUN-device" indicates that the device type identified by the ADARUN DEVICE parameter will be used as a default if a different device type is not specified. The literal "ADARUN-device" should *not* be specified for the SORTDEV parameter.

underlining	Underlined text identifies default values, wherever appropriate. Otherwise, the defaults are explained in the accompanying parameter descriptions.	[LRECL = { *record-buffer-length* | 4000 }] In the example above, 4000 is the default that will be used for the LRECL parameter if no other record buffer length is specified.

Convention	Description	Example
vertical bars (|)	Vertical bars are used to separate mutually exclusive choices. **Note:** In more complex syntax involving the use of large brackets or braces, mutually exclusive choices are stacked instead.	ADAORD { RESTRUCTUREF | REF } In the example above, you must select RESTRUCTUREF or REF for this ADAORD function. There are no defaults.
brackets ([ ])	Brackets are used to identify optional elements. When multiple elements are stacked or separated by vertical bars within brackets, only one of the elements may be supplied.	SORTSEQ = { descriptor [,MU] [,NU] | ISN [, STARTISN = value] | physical-sequence } In this example, the SORTSEQ parameter and the MU, NU, and STARTISN subparameters are optional. **Note:** Note that the mutually exclusive choices for the SORTSEQ parameter are stacked.

braces ({ })	Braces are used to identify required elements. When multiple elements are stacked or separated by vertical bars within braces, one and only one of the elements must be supplied.	SUSPEND
		    [TTSYN = {time-available-to-sync | ADARUN-TT }]
		    [TRESUME = {time-until-resume | 120 }]
		RESUME
		
		In this example, either the SUSPEND or RESUME parameter is required. If SUSPEND is specified, either or both TTSYN or TRESUME may be specified.

indentation	Indentation is used to identify subparameters of a parameter.	SUSPEND
		    [TTSYN = {time-available-to-sync | ADARUN-TT }]
		    [TRESUME = {time-until-resume | 120 }]
		RESUME
		
		In this example, TTSYN and TRESUME are subparameters of the SUSPEND parameter.

ellipsis (...)	Ellipses are used to identify elements that can be repeated. If the term preceding the ellipsis is an expression enclosed in square brackets or braces, the ellipsis applies to the entire bracketed expression.	[FIELD = 'field-name [, option]... '] ... In this example, the FIELD parameter can be repeated. In addition, more than one option can be associated with a field.

Convention	Description	Example
other punctuation and symbols	All other punctuation and symbols must be entered exactly as shown.	[FIELD = ' *field-name* [, *option*]... '] ... In this example, the single quotation marks must be specified around the field definitions and their associated options. In addition, options must be separated by commas.

Adabas Utility Syntax

Adabas utilities have the following format:

utility function parameter-list

where:

utility	is the name of the utility to be executed. Examples of utility names include: ADAORD ADADBS ADAINV
function	is the name of the specific utility operation to be executed. For example: ADAORD REORDATA ADADBS ADD ADAINV COUPLE Most single-function utilities (ADASEL, ADAULD, etc.) whose function is implicit have either no function value or an optional one.
parameter-list	is a list of parameters following the function. The syntax of the parameters is the same as the syntax used elsewhere in this documentation. For more information, read *Syntax Conventions*.