Trafodion Install Guide

Version 1.3.0, January 2016

Table of Contents

License Statement

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Version	Date
1.3.0	January, 2016

Chapter 1. About This Document

This guide describes how to use Trafodion for:

- **Product Environments**: This install allows you to store and query data using Trafodion, either via Trafodion clients (see Trafodion Installation Guide or via application code you write.

- **Build Environments**: This install allows you to build Trafodion from the Trafodion source code.

1.1. Intended Audience

This guide is intended for:

- **Product Environments**: Database administrators and support personnel who are installing a Trafodion database for application usage.
  - **Single-Node Install**: Typically used when you want to evaluate Trafodion.
  - **Cluster Install**: Typically used when you deploy Trafodion for application usage.

- **Developer Environments**: Developers working on enhancing the Trafodion source code.
  - **Single-Node Install**: Most Trafodion development is done in a single-node environment.
  - **Cluster Install**: Mostly used for Trafodion quality assurance, performance, and scale testing.

The installation instructions applies to a diverse set of platforms:

- **Virtual Machines**: Often used for evaluations and Trafodion development.

- **Cloud**: Used for Product Environments as well as for Developer Environments.

- **Bare Metal**: Used for Product Environments as well as for Developer Environments.

1.2. New and Changed Information

This is a new guide.

1.3. Notation Conventions

This list summarizes the notation conventions for syntax presentation in this manual.

Notation	Usage	Example	Notes
UPPERCASE LETTERS	Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required.	SELECT	
Italic Letters	Italic letters, regardless of font, indicate variable items that you supply. Items not enclosed in brackets are required.	*file-name*	
`Computer Type`	Computer type letters within text indicate case-sensitive keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required.	`myfile.sh`	

Bold Text	Bold text in an example indicates user input typed at the terminal.	`ENTER RUN CODE:`
		`?`**123**
		`CODE RECEIVED: 123.00`
			The user must press the Return key after typing the input.

[ ] Brackets	Brackets enclose optional syntax items.	DATETIME [*start-field* TO] *end-field*	A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example: `DROP SCHEMA` `schema` `[CASCADE]` `DROP SCHEMA` `schema` `[CASCADE	RESTRICT]`

Notation	Usage	Example	Notes
{ } Braces	Braces enclose required syntax items.	`FROM { grantee[, grantee]···}`	A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example: `INTERVAL { start-field TO end-field }` `{ single-field }` `INTERVAL { start-field TO end-field	single-field }`—
	Vertical Line	A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces.	`{expression	NULL}`	
... Ellipsis	An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times.	`ATTRIBUTE[S] attribute [, attribute]···` `{, sql-expression}···`	An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example: `expression-n···`
Punctuation	Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown.	DAY (*datetime-expression*)` `@script-file`	Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example: `"{" module-name [, module-name]··· "}"`
Item Spacing	Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma.	`DAY (datetime-expression)` `DAY(datetime-expression)`	If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items: `myfile.sh`

Notation	Usage	Example	Notes
Line Spacing	If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections.	`match-value` `[NOT]` `LIKE` `pattern` [ESCAPE *esc-char-expression*]	

1.4. Publishing History

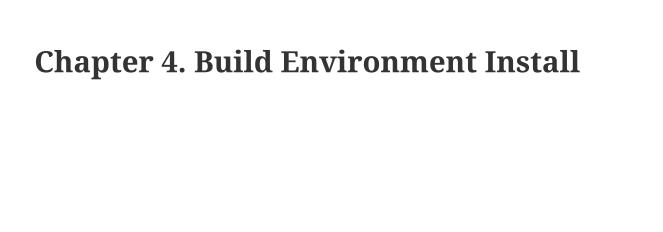| Product Version	Publication Date
Trafodion Release 1.3.0	To be announced.

1.5. Comments Encouraged

The Trafodion community encourages your comments concerning this document. We are committed to providing documentation that meets your needs. Send any errors found, suggestions for improvement, or compliments to:

issues@trafodion.incubator.apache.org

Include the document title and any comment, error found, or suggestion for improvement you have concerning this document.

`match-value` `[NOT]` `LIKE`
`pattern`

Chapter 2. Requirements

Chapter 3. Product Environment Install

Chapter 4. Build Environment Install

Chapter 5. Appendix

Chapter 6. How-To