

TIBCO BusinessWorks™

COBOL Copybook Plug-in

User's Guide

Software Release 2.5
October 2008

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN TIBCO BUSINESSWORKS COBOL COPYBOOK PLUG-IN USER'S GUIDE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIB, TIBCO, TIBCO Adapter, Predictive Business, Information Bus, The Power of Now, TIBCO Rendezvous, TIBCO InConcert, TIBCO Administrator, and TIBCO BusinessWorks are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README.TXT FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2004-2008 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Preface	v
Related Documentation	vi
TIBCO BusinessWorks Documentation	vi
Other Documentation	vi
How to Contact TIBCO Support	viii
 Chapter 1 Introduction	 1
Overview of the COBOL Copybook Palette	2
Installing the Software	4
Supported Platforms	4
Prerequisites	5
Installation	5
Examples of Using the COBOL Copybook Palette	6
Example of Converting a Copybook Schema to an XSD	7
COBOL Data Division Features	10
Supported COBOL Usages	11
Supported COBOL Picture Types	12
Supported COBOL Clauses and Features	13
Unsupported Clauses, Features, Levels, and Phrases	15
Handling of Zoned Decimal Items	16
 Chapter 2 The COBOL Copybook Palette	 19
Copybook Schema	20
Configuration	20
Import Copybook Schema Button	20
Copybook	20
XML Schema	21
Test Parsing Button	21
Configure REDEFINES	21
OCCURS ... DEPENDING ON Clause	26
Parse Copybook Data	29
Configuration	29
Input	31
Parsing a Large Number of Records	32
Output	33
Error Output	33

Render Copybook Data 34

 Configuration 34

 Input 37

 Output 37

 Error Output 37

Chapter 3 Error Codes 39

COBOL Copybook Error Codes 40

TIBCO Software Inc. End User License Agreement 41

Index 45

Preface



This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

TIBCO BusinessWorks is a standards-based, easy-to-deploy solution for companies looking to integrate their enterprise computing environment and automate their business processes. This manual describes the TIBCO BusinessWorks COBOL Copybook Plug-in; it is not intended to be a complete reference for developing process definitions using TIBCO BusinessWorks. Understanding the content of the TIBCO BusinessWorks documentation is a prerequisite to using this product. This manual is also not intended to be a complete reference for concepts or syntax of COBOL copybooks. Understanding of the COBOL Copybook specification is a prerequisite to using the TIBCO BusinessWorks COBOL Copybook Plug-in.

Topics

- [*Related Documentation, page vi*](#)
- [*How to Contact TIBCO Support, page viii*](#)

Related Documentation

You may find the following documentation resources useful.

TIBCO BusinessWorks Documentation

The following documents are part of the TIBCO BusinessWorks documentation set:

- *TIBCO BusinessWorks Concepts* Read this manual before reading any other manual in the documentation set. This manual describes terminology and concepts of TIBCO BusinessWorks, and the other manuals in the documentation set assume you are familiar with the information in this manual.
- *TIBCO BusinessWorks Quick Start* This manual steps you through a very simple example of designing, deploying, and monitoring a TIBCO BusinessWorks process.
- *TIBCO BusinessWorks Process Design Guide* This manual describes how to create, edit, and test business processes using TIBCO BusinessWorks.
- *TIBCO BusinessWorks Installation* Read this manual for information on installing one or more components of TIBCO BusinessWorks and setting up a TIBCO BusinessWorks domain.
- *TIBCO BusinessWorks Error Codes* This manual describes errors returned by TIBCO BusinessWorks.
- *TIBCO BusinessWorks Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closes issues for this release.
- *TIBCO BusinessWorks COBOL Copybook Plug-in User's Guide* This manual describes TIBCO BusinessWorks COBOL Copybook Plug-in.
- *TIBCO BusinessWorks COBOL Copybook Plug-in Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closes issues for this release.

Other Documentation

TIBCO BusinessWorks is bundled with other products. You will therefore find the documentation for those products useful:

- TIBCO Designer™ documentation. TIBCO Designer is an easy to use graphical user interface for design-time configuration of TIBCO applications. TIBCO Designer includes online help for each palette.
- TIBCO Administrator™ documentation. TIBCO Administrator is the monitoring and managing interface for new-generation TIBCO products such as TIBCO BusinessWorks.
- Information on COBOL copybook concepts and syntax is available from many sources, such as the IBM publication *Enterprise COBOL for z/OS V3.4 Language Reference*.

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

Chapter 1

Introduction

COBOL Copybook files describe the schema used for COBOL data files. COBOL Copybook files usually have the extension `.cpy`, `.cob`, or `.cb1` and are roughly analogous to include files in the C programming language. The COBOL Copybook palette allows you to interpret or create COBOL data based on the specified Copybook file.

Topics

- [*Overview of the COBOL Copybook Palette, page 2*](#)
- [*Installing the Software, page 4*](#)
- [*Examples of Using the COBOL Copybook Palette, page 6*](#)
- [*Example of Converting a Copybook Schema to an XSD, page 7*](#)
- [*COBOL Data Division Features, page 10*](#)

Overview of the COBOL Copybook Palette

The COBOL Copybook palette provides a mechanism for interacting with applications that produce COBOL data. The data schema of a COBOL program is defined within the COPY directive within the WORKING STORAGE section. You can store the COBOL Copybook into a file, and these files typically have a .cpy, .cob, or .cbl file extension. These files define the structure of the data that the COBOL program uses and they are roughly analogous to include files in the C programming language.

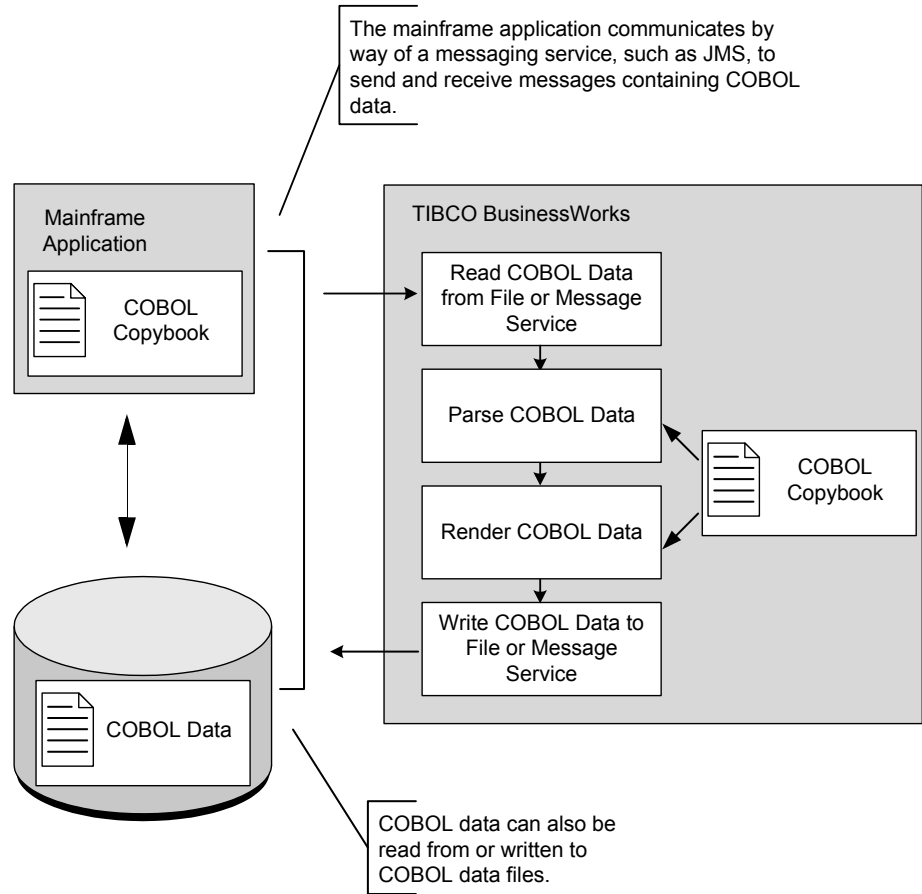
To use data produced by a COBOL program, TIBCO BusinessWorks must parse the data into XML so that activities within a process definition can access it. The COBOL Copybook palette has the following resources that allow you to manipulate COBOL data:

- [Copybook Schema](#) — This shared configuration resource specifies the schema contained within a Copybook file.
- [Parse Copybook Data](#) — This activity parses incoming COBOL data based on a specified Copybook Schema resource. After parsing, the data is contained within an XML schema and other activities can access the data.
- [Render Copybook Data](#) — This activity accepts XML data and produces COBOL data based on the specified Copybook schema. The formatted COBOL data can then be passed to a COBOL application.

Typically, COBOL applications communicate to other applications by way of a messaging service, such as IBM MQ Series or Java Messaging Service (JMS). COBOL applications can also read data from or write data to a file. TIBCO BusinessWorks can retrieve or send COBOL data by way of a messaging service or by using the file system.

[Figure 1](#) illustrates using the BusinessWorks COBOL Copybook Plug-in to receive and send COBOL data to and from a mainframe application or a COBOL data file. The COBOL Copybook schema is used to define the structure of the data.

Figure 1 Using the BusinessWorks COBOL Copybook Plug-in



Installing the Software

This section describes how to install the BusinessWorks COBOL Copybook Plug-in.

Supported Platforms

The BusinessWorks COBOL Copybook Plug-in is supported on the following platforms:

Microsoft Windows (x86)

- Windows 2000 (With Service Pack 4)
- Windows XP (Professional with Service Pack 2)
- Windows Server 2003
- Solaris 10

Solaris (SPARC)

- Solaris 7
- Solaris 8
- Solaris 9
- Solaris 10

HP-UX (PA-RISC)

- HP-UX 11.0
- HP-UX 11i

HP-UX (IA64 Itanium)

- HP-UX 11.23
- HP-UX 11i V2

Linux (x86)

- Linux Kernel 2.4.18

IBM (POWER)

- AIX 5.2
- AIX 5.3

Prerequisites

TIBCO BusinessWorks 5.4.0 or above must be installed on the system according to the instructions in *TIBCO ActiveMatrix BusinessWorks Installation*.

Installation

To install the product, perform the following:

1. Make sure that TIBCO BusinessWorks 5.4.0 or above is installed.
2. Exit TIBCO Designer, if it is open.
3. Insert the BusinessWorks COBOL Copybook Plug-in installation CD-ROM or download the software.
4. Navigate to the directory appropriate for your platform and then invoke the installer.

The installer asks you to accept the License Agreement and then installs the product.

Examples of Using the COBOL Copybook Palette

The COBOL Copybook palette can be used to either send data to or retrieve data from a COBOL application. This section describes examples of using the COBOL Copybook palette to accomplish either goal.

Figure 2 illustrates a process that receives a JMS queue message containing COBOL data. The ParseCopybookData activity is used to convert the data into XML, and the data is passed to the HandleIncomingData subprocess.

Figure 2 Receiving data from a COBOL Application

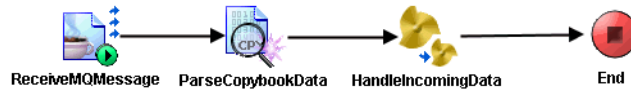
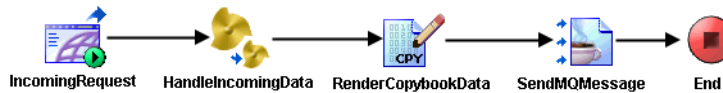


Figure 3 illustrates data received from an incoming HTTP request and then passed to the HandleIncomingData subprocess. The data is then passed to the RenderCopybookData activity so that COBOL data can be created to send to a COBOL application. The data is then sent to the COBOL application by way of a JMS Queue Sender activity.

Figure 3 Sending data to a COBOL application



The examples above illustrate the basic usage of parsing, rendering, sending, and receiving. Your integration project could be more complex and require any combination of parsing, rendering, sending, or receiving COBOL data in the same process definition.

Example of Converting a Copybook Schema to an XSD

The Copybook Schema resource converts a COBOL Copybook into an XSD that is used to parse COBOL data into an XML document and conversely render XML as COBOL data. The following COBOL Copybook is a simple example of a data schema for a purchase order.

```
01 PURCHASE-ORDER.
  02 HEADER.
    03 DOCUMENT-DEP PIC 9.
    03 DOCUMENT-DATE-TIME PIC X(20) VALUE IS SPACES.
  02 LINEITEM.
    03 ORDERITEM-DEP PIC 9.
    03 QUANTITY PIC 9(4) USAGE IS COMP-3.
    03 UNIT-PRICE.
      04 UNIT-PRICE-CONTROL PIC X.
      04 ACTUAL.
        05 CURRENCY PIC X(10) VALUE IS SPACES.
        05 FILLER PIC 99.
```

The example Copybook schema above is converted to the following XSD:

```
<xs:schema xmlns:cobol=http://xmlns.tibco.com/cobol2schema
xmlns:xs=http://www.w3.org/2001/XMLSchema
cobol:copybookType="COBOL" cobol:prefix=""
elementFormDefault="unqualified"
attributeFormDefault="unqualified">
  <xs:complexType name="_ROOT_TYPE">
    <xs:choice>
      <xs:element cobol:byteLength="38" cobol:byteOffset="0"
name="PURCHASE-ORDER" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element cobol:byteLength="21" cobol:byteOffset="0"
name="HEADER" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element cobol:byteLength="1"
cobol:picture="9"
cobol:fullName="PURCHASE-ORDER.HEADER.DOCUMENT-
DEP"
cobol:byteOffset="0" cobol:usage="DISPLAY"
name="DOCUMENT-DEP" minOccurs="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:unsignedShort">
                        <xs:totalDigits value="1"/>
                        <xs:minInclusive value="0"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element cobol:byteLength="20"
cobol:picture="X(20)"
cobol:fullName="PURCHASE-ORDER.HEADER.DOCUMENT-
DATE-TIME"
```

```

        cobol:byteOffset="1" cobol:usage="DISPLAY"
        name="DOCUMENT-DATE-TIME" default="SPACES"
        minOccurs="0">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="20"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element cobol:byteLength="17"
cobol:byteOffset="21" name="LINEITEM" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element cobol:byteLength="1"
            cobol:picture="9"
            cobol:fullName="PURCHASE-ORDER.LINEITEM.
            ORDERITEM-DEP"
            cobol:byteOffset="21" cobol:usage="DISPLAY"
            name="ORDERITEM-DEP" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base="xs:unsignedShort">
                        <xs:totalDigits value="1"/>
                        <xs:minInclusive value="0"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element cobol:byteLength="3"
            cobol:picture="9(4)"
            cobol:fullName="PURCHASE-ORDER.LINEITEM.
            QUANTITY"
            cobol:byteOffset="22"
            cobol:usage="PACKED-DECIMAL" name="QUANTITY"
            minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base="xs:unsignedShort">
                        <xs:totalDigits value="4"/>
                        <xs:minInclusive value="0"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element cobol:byteLength="13"
            cobol:byteOffset="25" name="UNIT-PRICE"
            minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element cobol:byteLength="1"
                        cobol:picture="X"
                        cobol:fullName="PURCHASE-ORDER.LINEITEM.
                        UNIT-PRICE.UNIT-PRICE-CONTROL"
                        cobol:byteOffset="25"
                        cobol:usage="DISPLAY"
                        name="UNIT-PRICE-CONTROL" minOccurs="0">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">

```

```

        <xs:maxLength value="1"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element cobol:byteLength="12"
cobol:byteOffset="26"
name="ACTUAL" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element cobol:byteLength="10"
cobol:picture="X(10)"
cobol:fullName="PURCHASE-ORDER.LINE
ITEM.UNIT-PRICE.ACTUAL.CURRENCY"
cobol:byteOffset="26"
cobol:usage="DISPLAY"
name="CURRENCY"
default="SPACES" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base=
"xs:string">
                        <xs:maxLength value="10"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element cobol:byteLength="2"
cobol:picture="99"
cobol:fullName="PURCHASE-ORDER.LINE
ITEM.UNIT-PRICE.ACTUAL.FILLER"
cobol:byteOffset="36"
cobol:usage="DISPLAY"
name="FILLER" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base=
"xs:unsignedShort">
                        <xs:totalDigits value="2"/>
                        <xs:minInclusive value="0"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:schema>

```

COBOL Data Division Features

The section lists COBOL data division features that the BusinessWorks COBOL Copybook Plug-in supports and the features it does not support. The information is organized as follows:

- [Supported COBOL Usages](#)
- [Supported COBOL Picture Types](#)
- [Supported COBOL Clauses and Features](#)
- [Unsupported Clauses, Features, Levels, and Phrases](#)
- [Handling of Zoned Decimal Items](#)



This release of BusinessWorks COBOL Copybook Plug-in introduces new features and functionality. For a complete, detailed list, see *TIBCO BusinessWorks COBOL Copybook Plug-in Release Notes*. During migration, the new features may cause the plug-in to work differently than expected. To ensure the integrity of your data, please note the following recommendation.

If you use TIBCO BusinessWorks processes defined using a previous version of the plug-in:

Install the plug-in and test the processes. If the results are as expected, then migrate the processes to the new version of the plug-in.

Supported COBOL Usages

Table 1 Supported COBOL Usages

Usage	Notes
BINARY	See COMP Clause Usage on page 13.
COMP, COMPUTATIONAL	
COMP-4, COMPUTATIONAL-4	
COMP-5, COMPUTATIONAL-5	
COMP-3, COMPUTATIONAL-3	
PACKED-DECIMAL	
DISPLAY	
COMP-1, COMPUTATIONAL-1	The data element with this usage is converted to the <code>xs:float</code> type in the output XSD schema.
COMP-2, COMPUTATIONAL-2	<p>The data element with this usage is converted to the <code>xs:double</code> type in the output XSD schema.</p> <p>Note: Only IEEE 754 representation is supported for the COMP-1 and COMP-2 fields. IBM Hexadecimal Floating point is not supported and will yield incorrect results.</p>
FUNCTION-POINTER	The data element with this usage is converted to the <code>xs:integer</code> type in the output XSD schema.
INDEX	
POINTER	
PROCEDURE-POINTER	The data element with this usage is converted to the <code>xs:long</code> type in output XSD schema.

Supported COBOL Picture Types

Table 2 Supported COBOL Picture Types

Picture Type	Example	Notes
Alphabetic	PIC AAA PIC A(12)	The data element with this picture type is converted to the <code>xs:string</code> type in the output XSD schema.
Alphanumeric	PIC XX PIC X(10)	
Alphanumeric-edited	PIC XX/XX/XX PIC XXXBX(3)BX(4)	
Numeric-edited	PIC 99/99/99 PIC -9(7).99 PIC +,+++,999.00	
External Floating Point	PIC +99.9999E+99 PIC -9.99999E-99	The data element with this picture type is converted to the <code>xs:double</code> type in the output XSD schema.
Numeric	PIC 999 PIC 9(5) PIC S9(8) PIC S9(5)V99	<p>If the picture for a data element has a decimal value, the clause is converted to the <code>xs:decimal</code> type in the output XSD schema.</p> <p>If the picture for a data element does not have any decimal value, the data element is converted to the signed or unsigned integer or short type in the output XSD schema.</p> <p>Whether the converted-to type is signed or unsigned integer is decided by the presence of sign character in the picture.</p> <p>If the number of digits in the picture is 5 or more, the data element is converted to integer type. Otherwise, it is converted to short.</p>

Supported COBOL Clauses and Features

Table 3 Supported COBOL Clauses and Features

Clause/Feature	Notes
OCCURS	The data element defined as a fixed length table using the OCCURS clause.
OCCURS ... DEPENDING ON	The data element defined as a variable length table using the OCCURS ... DEPENDING ON clause.
REDEFINES	The data elements that are defined using the REDEFINES clause are converted to the <code>xs:choice</code> type in the output XSD schema.
SIGN	The data element defined using the SIGN clause is supported.
SYNCHRONIZED	None.
Multiple Data Element Definitions in Copybook	The copybook supports multiple data element definitions. Within a schema, each data element definition is treated as a group element when you convert the copybook to XSD.
VALUE	The VALUE clause is reflected in the XSD schema and the Render Copybook Data activity will generate that value for items that do not have values supplied in the input data.

COMP Clause Usage

Numeric values in COBOL files are commonly stored in display or character format. That is, the value is stored as a base-10 number, with each digit represented by the corresponding character. For example, a field defined as PIC 999 containing the value 123 is stored in three bytes, each byte containing one digit of the value.

When performing computations with numbers, machines can perform the computations significantly faster on binary (base-2) numbers rather than base-10 numbers. Therefore, if a number is stored in a COBOL data file in binary format, it can be input directly from the file and used in computations. Various flavours of usage COMP (such as COMP, COMP-4, and so on), are used to change the storage format from text to a binary form.

COMP versus COMP-5

In COBOL, for integer fields, usage COMP is used to specify a storage of half word, full word, or double word (2, 4, and 8 bytes, respectively, big-endian representation). However, for COMP usage there is an additional limitation based on the number of decimal digits in the PICTURE clause. For instance PIC S9(4) COMP will have signed half word storage, but it will limit the values to the range of -9999 though +9999 due to 4 decimal places in the PICTURE clause.

In order to remove the above limitation, COMP-5 can be used. This usage is the same as COMP, but it allows the whole range of values representable by a particular number of storage bytes. For example PIC S9(4) COMP-5 has a half-word storage and allows the range of values of -32768 though +32767.

See COBOL documentation for more details.

Unsupported Clauses, Features, Levels, and Phrases

Table 4 *Unsupported Clauses, Features, Levels, and Phrases*

Clause/Feature/Level/Phrase	Notes
66	The data elements defined using the level 66 (RENAMES clause) are ignored.
88	The data elements defined using the level 88 (VALUE or VALUES clause) are ignored.
BLANK WHEN ZERO	This clause is ignored when the copybook is converted to the XSD schema.
DATE FORMAT	
EXTERNAL	
GLOBAL	
JUSTIFIED	
Nested Copybooks	Copybooks that contain the COPY directive for other copybooks are not supported.
OCCURS Clause <ul style="list-style-type: none"> – ASCENDING – DESCENDING – INDEXED – KEY 	These phrases for the OCCURS clause are ignored.
SYNCHRONIZED Clause <ul style="list-style-type: none"> – LEFT – RIGHT 	These phrases for the SYNCHRONIZED clause are ignored.
USAGE Clause <ul style="list-style-type: none"> – NATIVE – OBJECT REFERENCE 	These phrases for the USAGE clause are ignored.

Handling of Zoned Decimal Items

Items with usage DISPLAY and a numeric PICTURE clause are treated as zoned decimal data unless SIGN SEPARATE is specified. This format uses textual representation for all decimal digits except the last or first byte (depending on LEADING or TRAILING in SIGN clause). This byte contains both the decimal value and the sign for the whole value. The sign is specified in the first nibble (half byte) and the decimal digit in the second nibble.

There are three variations of the zoned decimal format – EBCDIC, Strict ASCII, and Modified ASCII. The BusinessWorks COBOL Copybook Plug-in expects a particular format depending on the Character Set configuration value that is specified for the Render Copybook Data or Parse Copybook Data activity. For details on the configuration of these activities, see [Chapter 2](#).

EBCDIC Character Sets (such as Cp037)

- The Parse Copybook Data activity expects EBCDIC zoned decimal format and fails if any of the ASCII format data is encountered. EBCDIC zoned decimal format uses EBCDIC codepoints for digits and defines the following sign nibbles for the sign byte: xC and xA – sign positive, xD and xB – sign negative, xF – unsigned value. Positive and negative nibbles are used with signed items (with S in the PICTURE clause), xF is used for unsigned items (no S in the PICTURE). Note that the unsigned nibble is not valid for signed items and vice versa – the signed nibbles are not valid in an unsigned item.
- The Render Copybook Data activity will generate signed nibbles xC and xD for positive and negative signed values, respectively. For unsigned data, xF will be generated.

ASCII Character Sets

- The Parse Copybook Data will accept both Strict ASCII and Modified ASCII. Both formats use ASCII codepoints for decimal digits, but the sign byte is represented differently. For Strict ASCII, a nibble value x3 denotes positive and x7 negative value for the entire item. Note that there is no special value for unsigned data. Modified ASCII does not have predefined nibbles, see [Table 5, Modified ASCII Sign Byte](#).
- The Render Copybook Data activity will generate Strict ASCII format.



If the input data for the Parse Copybook Data activity does not conform to the expected format, an error will occur during execution.

If SIGN SEPARATE is specified, the data is represented as text in the selected character set with characters '+' or '-'.

Table 5 Modified ASCII Sign Byte

Sign Byte	Sign of Entire Value	Decimal Digit in Sign Byte
0x7B	positive	0
0x41	positive	1
0x42	positive	2
0x43	positive	3
0x44	positive	4
0x45	positive	5
0x46	positive	6
0x47	positive	7
0x48	positive	8
0x49	positive	9
0x7D	negative	0
0x4A	negative	1
0x4B	negative	2
0x4C	negative	3
0x4D	negative	4
0x4E	negative	5
0x4F	negative	6
0x50	negative	7
0x51	negative	8
0x52	negative	9

Chapter 2

The COBOL Copybook Palette

This chapter describes the resources of the COBOL Copybook palette.

Topics

- [*Copybook Schema, page 20*](#)
- [*Parse Copybook Data, page 29*](#)
- [*Render Copybook Data, page 34*](#)

Copybook Schema

Shared Configuration



The Copybook Schema shared configuration resource describes the schema to use when parsing or rendering COBOL data. This resource allows you to load a COBOL Copybook file by using the Import Copybook Schema button to choose a file containing the schema. These files typically have a .cpy, .cob, or .cbl extension.

Configuration

The configuration tab has the following fields.

Field	Global Var	Description
Name	No	The name to appear as the label for the resource.
COBOL Fixed Format?	No	Specifies whether the data is in COBOL fixed or free format. When checked, the data is expected to be in COBOL fixed format where columns 0-6 are reserved for line numbers. When unchecked, the data can be in COBOL free format where the COBOL code can be within columns 0-6.
Character Set	No	Specifies the character set encoding of the data.

Import Copybook Schema Button

The Import Copybook Schema button brings up a dialog that allows you to choose the file containing the COBOL Copybook schema you wish to import. Once the file is chosen, the COBOL schema appears in the Copybook tab and the XSD schema appears in the XML Schema tab.

Copybook

The Copybook tab displays the COBOL Copybook schema loaded into this resource.

XML Schema

The XML Schema tab displays the XSD that will be used to parse and render COBOL Copybook data.

Test Parsing Button

This button tests the COBOL Copybook schema and returns the results. When an error is encountered, the line number and column number of the error are displayed.

Configure REDEFINES

The COBOL REDEFINES clause allows different data description entries to describe the same storage area, similar to a union in most programming languages. The data description entries for REDEFINES are converted to a choice element in the XSD.

After you import the Copybook schema, you must configure the data description entries for the REDEFINES group. The configuration helps determine the unique data description entry for the group.

To configure the REDEFINES group, use one of the following methods:

- Assign a data description entry from the REDEFINES group as the default entry. The setting maps the data content of storage area to this default data description entry at run-time.
- Assign a data description entry (known as control field) to the REDEFINES group. At run-time, the value of this control field determines the data description entry to be used to map the data content of redefined storage area.

If a REDEFINES group is not configured at all (that is, neither a default data description entry nor a control field is assigned), at runtime, the first data description entry from the REDEFINES group is used to map data content of the redefined storage area.

For example, the following Copybook has a control field named UNIT-PRICE-CONTROL that determines whether the next field is ACTUAL or CONVERTED.

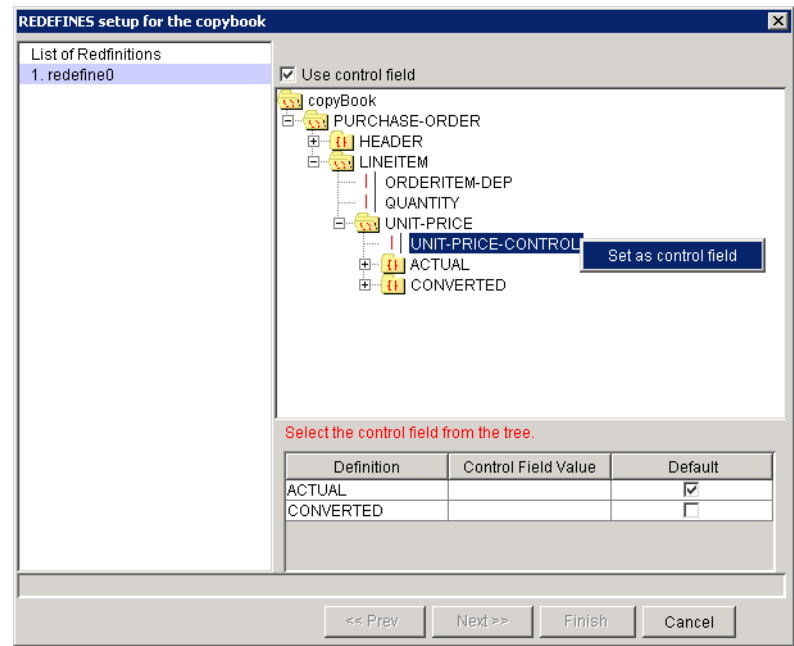
```
01 PURCHASE-ORDER.
  02 HEADER.
    03 DOCUMENT-DEP PIC 9.
    03 DOCUMENT-DATE-TIME PIC X(20) VALUE IS SPACES.
  02 LINEITEM.
    03 ORDERITEM-DEP PIC 9.
    03 QUANTITY PIC 9(4).
    03 UNIT-PRICE.
```

```
04 UNIT-PRICE-CONTROL PIC X.  
04 ACTUAL.  
    05 CURRENCY PIC X(10) VALUE IS SPACES.  
    05 FILLER PIC 99.  
04 CONVERTED REDEFINES ACTUAL.  
    05 CURRENCY PIC X(10).  
    05 CONVERSION-FACTOR PIC 99.
```

The Configure REDEFINES button brings up a dialog that allows you to specify how the contents of the control field determine which data element occurs. To use this dialog, follow this procedure:

- 1. Click the Configure REDEFINES button and select the REDEFINES clause from the list on the left that you wish to configure. There may be more than one REDEFINES clause in your COBOL Copybook schema, and this dialog lets you configure them individually.
- 2. Make sure the Use control field checkbox is checked. This checkbox is located above the list on the right.
- 3. Select the field that you would like to use as the control field.
- 4. Right-click on the control field and select Set as control field from the popup menu. [Figure 4](#) illustrates selecting the control field.

Figure 4 Setting the control field

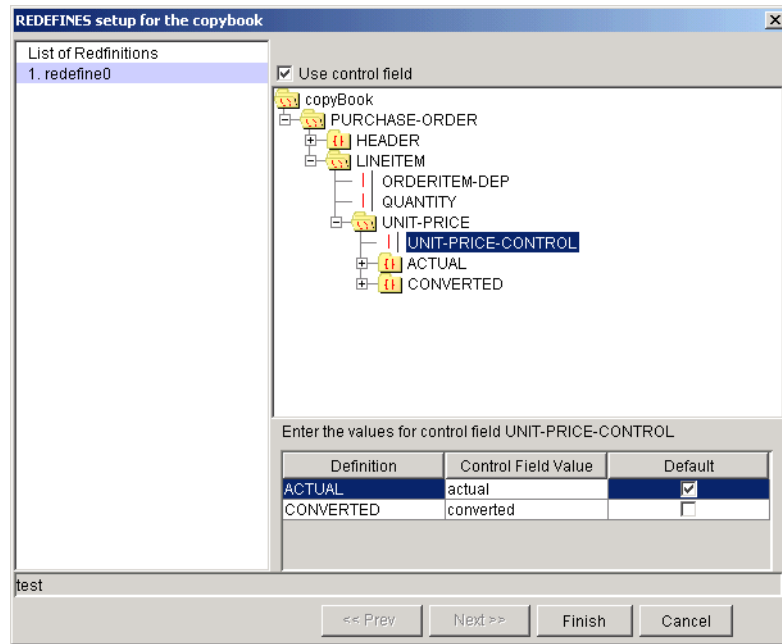


5. Select each row in the Control Field Value column of the table below the schema tree. Enter the value in the control field that signifies the data element. For example, if the UNIT-PRICE-CONTROL contains the value "actual", then ACTUAL is the data element. If the field contains "converted", then CONVERTED is the data element. Enter the correct value for each data element.



- You can enter multiple values in the Control Field Value column if more than one value can be used. For example, you may enter actual;act;a if the ACTUAL data element should be selected when the control field value is actual or act or a. Use a semicolon (;) to separate each potential value.
 - For text control fields, you can enter values that have whitespaces in them. During execution, these values will be processed based on the Parse Activity Trim Whitespaces from String Values. If checked, whitespaces will be trimmed from strings in the data as well as control field values (the spaces in the control field values will not be significant). When Trim Whitespaces from String Values is unchecked, then the unchanged values from data and unchanged control field values will be used for comparison (the spaces in the control field values will be significant). For additional details on Trim Whitespaces, see [Parse Copybook Data on page 29](#).
 - For control fields that are converted to numeric types, the control field values are being checked in a typed manner. For example, a control value '0003' will match data value 3 for fields converted to integer type, and '3E4' will match with '0.3E5' for float and double types.
6. Check the Default checkbox to assign one of the data elements to be the default element when the control field value matches none of the specified values. The default is the first data element in the redefines group.
 7. Click the Finish button. [Figure 5](#) illustrates the completed REDEFINES configuration for the example COBOL Copybook schema.

Figure 5 Completed REDEFINES configuration



The configuration of the REDEFINES clause in the example above would result in the following choice element in the XSD:

```
<xs:choice>
  <xs:element cobol:byteLength="12" cobol:isDefault="true"
    cobol:controlFieldValue="actual" cobol:byteOffset="27"
    cobol:controlField="PURCHASE-ORDER.LINEITEM.UNIT-PRICE.UNIT-
      PRICE-CONTROL"
    name="ACTUAL" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:element cobol:byteLength="10" cobol:picture="X(10)"
          cobol:fullName="PURCHASE-ORDER.LINEITEM.UNIT-PRICE.
            ACTUAL.CURRENCY"
          cobol:byteOffset="27" cobol:usage="DISPLAY"
          name="CURRENCY" default="SPACES" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="10"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element cobol:byteLength="2" cobol:picture="99"
          cobol:fullName="PURCHASE-ORDER.LINEITEM.UNIT-PRICE.
            ACTUAL.FILLER"
          cobol:byteOffset="37" cobol:usage="DISPLAY"
          name="FILLER" minOccurs="0">
          <xs:simpleType>
```

```

        <xs:restriction base="xs:unsignedShort">
            <xs:totalDigits value="2"/>
            <xs:minInclusive value="0"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element cobol:byteLength="12"
cobol:controlFieldValue="converted" cobol:byteOffset="27"
cobol:controlField="PURCHASE-ORDER.LINEITEM.UNIT-PRICE.UNIT-
PRICE-CONTROL"
name="CONVERTED" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element cobol:byteLength="10" cobol:picture="X(10)"
cobol:fullName="PURCHASE-ORDER.LINEITEM.UNIT-PRICE.
CONVERTED.CURRENCY"
cobol:byteOffset="27" cobol:usage="DISPLAY"
name="CURRENCY" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:maxLength value="10"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element cobol:byteLength="2" cobol:picture="99"
cobol:fullName="PURCHASE-ORDER.LINEITEM.UNIT-PRICE.
CONVERTED.CONVERSION-FACTOR"
cobol:byteOffset="37" cobol:usage="DISPLAY"
name="CONVERSION-FACTOR" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base="xs:unsignedShort">
                        <xs:totalDigits value="2"/>
                        <xs:minInclusive value="0"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:choice>

```



If the control field is a repeating field, then the value used to determine the actual choice is the last value processed before the `xs:choice` element.

OCCURS ... DEPENDING ON Clause

The COBOL OCCURS ... DEPENDING ON clause defines a repeating data element whose number of occurrences depends upon a numeric element. This type of element is automatically converted to a repeating element in the XSD.

For example, the following Copybook contains an element named ORDERITEM that occurs between 0 and 10 times. The number of times the element occurs is determined by the ORDERITEM-DEP field.

```
01 PURCHASE-ORDER.
  02 HEADER.
    03 DOCUMENT-DEP PIC 9.
    03 DOCUMENT-DATE-TIME PIC X(20) VALUE IS SPACES.
  02 LINEITEM.
    03 ORDERITEM-DEP PIC 9.
    03 ORDERITEM OCCURS 0 TO 10 TIMES DEPENDING ON ORDERITEM-DEP.
      04 ITEM-TYPE PIC X(10) VALUE IS SPACES.
      04 DESCR PIC X(40) VALUE IS SPACES.
      04 SERIAL-NUM PIC 9999.
    03 QUANTITY PIC 9(4).
    03 UNIT-PRICE.
      04 UNIT-PRICE-CONTROL PIC X.
```

The XSD element for the LINEITEM element would be the following:

```
<xs:element cobol:byteLength="546" cobol:byteOffset="21"
name="LINEITEM" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element cobol:byteLength="1" cobol:picture="9"
cobol:fullName="PURCHASE-ORDER.LINEITEM.ORDERITEM-DEP"
cobol:byteOffset="21" cobol:usage="DISPLAY"
name="ORDERITEM-DEP" minOccurs="0">
        <xs:simpleType>
          xs:restriction base="xs:unsignedShort">
            <xs:totalDigits value="1"/>
            <xs:minInclusive value="0"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element cobol:byteLength="54" cobol:occursMin="0"
cobol:occursMax="10" cobol:byteOffset="22"
cobol:occursDepending="PURCHASE-ORDER.LINEITEM.ORDERITEM-
DEP"
name="ORDERITEM" minOccurs="0" maxOccurs="10">
        <xs:complexType>
          <xs:sequence>
            <xs:element cobol:byteLength="10"
cobol:picture="X(10)"
cobol:fullName="PURCHASE-ORDER.LINEITEM.ORDERITEM.
ITEM-TYPE"
cobol:byteOffset="22" cobol:usage="DISPLAY"
name="ITEM-TYPE" default="SPACES" minOccurs="0">
              <xs:simpleType>
                <xs:restriction base="xs:string">
```

```

        <xs:maxLength value="10"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element cobol:byteLength="40"
cobol:picture="X(40)"
cobol:fullName="PURCHASE-ORDER.LINEITEM.ORDERITEM.
DESCR"
cobol:byteOffset="32" cobol:usage="DISPLAY"
name="DESCR" default="SPACES" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="40"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element cobol:byteLength="4" cobol:picture="9999"
cobol:fullName="PURCHASE-ORDER.LINEITEM.ORDERITEM.
SERIAL-NUM"
cobol:byteOffset="72" cobol:usage="DISPLAY"
name="SERIAL-NUM" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedShort">
        <xs:totalDigits value="4"/>
        <xs:minInclusive value="0"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element cobol:byteLength="4" cobol:picture="9(4)"
cobol:fullName="PURCHASE-ORDER.LINEITEM.QUANTITY"
cobol:byteOffset="-1" cobol:usage="DISPLAY" name="QUANTITY"
minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:unsignedShort">
      <xs:totalDigits value="4"/>
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element cobol:byteLength="1" cobol:byteOffset="-1"
name="UNIT-PRICE" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element cobol:byteLength="1" cobol:picture="X"
cobol:fullName="PURCHASE-ORDER.LINEITEM.UNIT-PRICE.
UNIT-PRICE-CONTROL"
cobol:byteOffset="-1" cobol:usage="DISPLAY"
name="UNIT-PRICE-CONTROL" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```



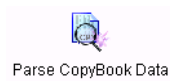
The DEPENDING ON clause object that defines the actual number of occurrences does not have to be of either integer or short type. The number of occurrences will be determined without errors if the actual value (after parsing or before rendering) can be converted to an integer. For example, a PIC X(4) field containing the value '0003' will mean 3 occurrences. If the value cannot be converted to a number, than parsing fails.

Whitespaces are always trimmed prior to the further processing. For example, a PIC X(4) field with the value '03' yields 3 occurrences.

If a field used in the OCCURS ... DEPENDING ON clause is a repeating field, the value that will be used for determining the actual number of occurrences is the last value processed before this OCCURS ... DEPENDING ON clause.

Parse Copybook Data

Activity



Parse CopyBook Data

The Parse Copybook Data activity parses incoming COBOL data based on the schema defined in the [Copybook Schema](#) resource. The data is converted to XML and can then be used by other activities within a process definition.

Configuration

The configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Copybook Schema Resource	No	The Copybook Schema shared configuration resource that defines the schema to use to parse the incoming COBOL data into XML.
Are the binary (COMP) numeric values in Big Endian format?	No	This field specifies the byte order of the incoming data. Check this field to specify the byte order is Big Endian (for example, this format is used on IBM mainframes and some UNIX platforms). Uncheck this field if the byte order is Little Endian (for example, this format is used by MS Windows and some UNIX platforms).
Character Set	No	<p>The character set encoding used by string values in the incoming data; also affects the parsing of zoned decimal values. For details, see Handling of Zoned Decimal Items on page 16.</p> <p>Note that only single byte character sets are available in the Parse Copybook Data activity.</p>

Field	Global Var?	Description
Trim Whitespaces from String Values	No	<p>Check this field to specify that the values converted to text type should have whitespace characters trimmed after parsing. Whitespace characters include:</p> <p>'\u0009' – HORIZONTAL TABULATION '\u000A' – LINE FEED '\u000B' – VERTICAL TABULATION '\u000C' – FORM FEED '\u000D' – CARRIAGE RETURN '\u001C' – FILE SEPARATOR '\u001D' – GROUP SEPARATOR '\u001E' – RECORD SEPARATOR '\u001F' – UNIT SEPARATOR '\u0000' – NULL CHARACTER</p> <p>Note: internally a trim() method of Java class String is used.</p> <p>This field also affects the how the redefine control field values are matched. For details, see Configure REDEFINES on page 21.</p>
Multiple Records	No	<p>Check this field's checkbox to specify that there is more than one record in the input data.</p>
Records Delimiter	No	<p>Specifies the delimiter used between records in the input data. The following are the types of delimiters you can specify:</p> <p>None — specifies that a new record begins directly after the last record ends. There is no character separating each record.</p> <p>New Line — specifies that a new line character separates records in the input.</p> <p>Carriage Return — specifies that a carriage return character separates records in the input.</p> <p>Carriage Return/Line Feed — specifies that a carriage return character followed by a line feed character separates records in the input.</p>

Field	Global Var?	Description
Input Type	No	Specifies the type of input for this activity. Input can either be a byte array or a file. If the input is a byte array, supply the string to the <code>bytes</code> input item. If the input is a file, supply the file name and location to the <code>fileName</code> input item.

Input

See *TIBCO BusinessWorks Process Design Guide* for more information about mapping and transforming input data.

The input for the activity is the following.

Input Item	Datatype	Description
bytes	binary	A byte stream containing COBOL data. If your COBOL data is a text string, you must convert the text to bytes before passing the input data to this activity. You can use the <code>string-to-base64()</code> XPath function to convert text to a byte array. This input item is only available when "Bytes" is specified in the Input Type field on the Configuration tab.
fileName	string	The location and name of the file to read. The file's contents are used as the input for this activity. This input item is only available when "File" is specified in the Input Type field on the Configuration tab.

Input Item	Datatype	Description
noOfRecords	number	<p>This element is only available when File is chosen in the Input Type field on the Configuration tab. This element specifies the number of records to read from the input stream. The default value of this element is "-1" which reads all records in the input stream.</p> <p>This is useful if you wish to read the input stream in parts to minimize memory usage. For details, see Parsing a Large Number of Records.</p>

Parsing a Large Number of Records

The input for this activity is placed in a process variable and takes up memory as it is being processed. If you are reading a large number of records from a file, the process may consume significant machine resources. To avoid using too much memory, you may wish to read the input in parts, parsing and processing a small set of records before moving on to the next set.

To process a large number of records, follow this procedure:

1. Create a Parse Copybook Data activity.
2. Specify the fields on the Configuration tab. Select file in the Input Type field.
3. Select the Parse Copybook Data activity and click the group icon on the toolbar to create a group containing the Parse Copybook Data activity. See *TIBCO BusinessWorks Process Design Guide* for more information about working with groups.
4. Specify "Repeat Until True Loop" as the Group action, and specify an index name (for example, "i").
5. The loop should exit when the EOF output element for the Parse Copybook Data activity is set to true. For example, the condition for the loop could be set to the following:
`string($ParseCopybookData/Output/EOF) = string(true())`
6. Set the noOfRecords input item for the Parse Copybook Data activity to the number of records you wish to process for each execution of the loop.
7. The loop processes the specified noOfRecords with each iteration until there are no more input records to parse.

The procedure above is a general guideline for creating a loop group for parsing a large set of input records in parts. You may want to modify the procedure to include additional processing of the records, or you may want to change the XPath expressions to suit your business process.

Output

The output for the activity is the following.

Output Item	Datatype	Description
<CopybookSchema>	complex	The parsed data represented as an XML schema.
numOfRecords	integer	Number of records in the parsed XML data.
EOF	boolean	<p>true if no more records are available for parsing. false if there are more records available.</p> <p>This output item is useful for checking if there are no more records in the input stream when you are reading the input in parts to preserve memory. See Parsing a Large Number of Records on page 32 for more information.</p>

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. See *TIBCO BusinessWorks Error Codes* for more information about error codes and corrective action to take.

Exception	Thrown When...
DeSerializationException	An error occurred when deserializing the incoming data.

Render Copybook Data

Activity



The Render Copybook Data activity allows you to supply XML data based on the schema defined by a COBOL Copybook resource and then renders that data as a byte stream that can be consumed by a COBOL program.

Configuration

The configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Copybook Schema Resource	No	The Copybook Schema shared configuration resource that defines the schema to use to render the XML data as COBOL data.
Are the binary (COMP) values in Big Endian format?	No	This field specifies the byte order of the rendered data. Check this field to specify the byte order is Big Endian (for example, this format is used on IBM mainframes and some UNIX platforms). Uncheck this field if the byte order is Little Endian (for example, this format is used by MS Windows and some UNIX platforms).
Character Set	No	<p>The character set encoding used by string values in the rendered data; also affects the rendering of zoned decimal values. For details, see Handling of Zoned Decimal Items on page 16.</p> <p>Note that only single byte character sets are available in the Render Copybook Data activity.</p>

Field	Global Var?	Description
Multiple Records?	No	<p>Uncheck if only one only record should be rendered. In this case, only the first child of the “Input” element from the input XML will be rendered as a single record. Otherwise, all child elements will be rendered as records, separated by the delimiter specified in the Records Delimiter field.</p>
Records Delimiter	No	<p>Specifies the delimiter used between records in the output data. The following are the types of delimiters you can specify:</p> <p>None — specifies that a new record begins directly after the last record ends. There is no character separating each record.</p> <p>New Line — specifies that a new line character separates records in the output.</p> <p>Carriage Return — specifies that a carriage return character separates records in the output.</p> <p>Carriage Return/Line Feed — specifies that a carriage return character followed by a line feed character separates records in the output.</p> <p>The above values are specific to the Unix and Windows platforms. Using record delimiters is not recommended when the Render As property is set to String. See the warning below.</p>
Filler Character	No	<p>Specifies the fill character to use between elements of each data record. NULL is the default fill character, but you can select either NULL or Space as the fill character.</p>

Field	Global Var?	Description
Render As	No	<p>Specifies the output data format. The output data can be rendered as one of the following:</p> <p>Bytes —The data is rendered as a byte stream.</p> <p>String —The data is rendered as a string. The data string has character set encoding as specified in the description of Character Set field in this table.</p> <p>Use with caution, see the warning below.</p>



You should use caution when specifying the String value for the Render As property. This value instructs the Render activity to generate a string for the entire result of rendering using a specific character set.

This is strongly discouraged when the copybook contains fields with binary representation, such as fields with usage BINARY (COMP), COMP-1 COMP-2, COMP3, or zoned decimal fields without SIGN SEPARATE. Refer to the COBOL specification for details.

In addition, Render As String is not recommended for rendering multiple records with a record delimiter. Possible values of the Records Delimiter property are specific to the Unix/Windows platforms and, depending on the character set, may yield unexpected characters.

When Render As String is used in the above cases, it is likely that the binary data will be converted to meaningless characters in the output string. Also, it is possible that the binary values will not be recoverable from the resulting string.

Input

See *TIBCO BusinessWorks Process Design Guide* for more information about mapping and transforming input data.

The input for the activity is the following.

Input Item	Datatype	Description
Input	complex	The XML schema defined by the Copybook Schema resource specified on the Configuration tab.

Output

The output for the activity is the following.

Output Item	Datatype	Description
bytes	binary	A byte stream containing COBOL data.
numOfRecords	integer	Number of records in the output byte stream.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. See *TIBCO BusinessWorks Error Codes* for more information about error codes and corrective action to take.

Exception	Thrown When...
SerializationException	An error occurred when serializing the data.

Chapter 3 **Error Codes**

This chapter describes error codes that can be returned by activities within the COBOL Copybook palette.

Topics

- [*COBOL Copybook Error Codes, page 40*](#)

COBOL Copybook Error Codes

BW-COBOL-100001: Error rendering the data.For more information, please check the console or 'Error Output'.

Role: errorRole

Category: BW_Plugin

Description: Error rendering the data.For more information, please check the console or 'Error Output'.

Resolution: Please Contact TIBCO Support

BW-COBOL-100002: Error parsing the data.For more information, please check the console or 'Error Output'.

Role: errorRole

Category: BW_Plugin

Description: Error parsing the data.For more information, please check the console or 'Error Output'.

Resolution: Please Contact TIBCO Support

BW-COBOL-100003: COBOL metadata loading error: Copybook schema %1 not found or not valid.

Role: errorRole

Category: BW_Plugin

Description: COBOL metadata loading error: Copybook schema %1 not found or not valid.

Resolution: Please Contact TIBCO Support

TIBCO Software Inc. End User License Agreement

READ THIS END USER LICENSE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THE SOFTWARE, YOU AGREE TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS, DO NOT DOWNLOAD OR INSTALL THE SOFTWARE AND RETURN IT TO THE VENDOR FROM WHICH IT WAS PURCHASED.

Upon your acceptance as indicated above, the following shall govern your use of the Software except to the extent all or any portion of the Software (a) is subject to a separate written agreement, or (b) is provided by a third party under the terms set forth in an Addenda at the end of this Agreement, in which case the terms of such addenda shall control over inconsistent terms with regard to such portion(s).

License Grant. The Software is the property of TIBCO or its licensors and is protected by copyright and other laws. While TIBCO continues to own the Software, TIBCO hereby grants to Customer a limited, non-transferable, non-exclusive, license to use the Number of Units set forth in the Ordering Document solely for Customer's internal business use.

License Term. The term of each license for the Software shall be either perpetual or on a term limited basis as set forth in the Ordering Document. If licensed on a term limited basis, the term shall be for a period of three years commencing on the Purchase Date, and on expiration, Customer must cease using and return or destroy all copies of the Software.

Restrictions. Customer agrees not to (a) make more copies than the Number of Units (except for a reasonable number of copies for archival and disaster recovery purposes) or use any unlicensed versions of the Software; (b) provide access to the Software to anyone other than employees, contractors, or consultants under written contract with Customer agreeing to be bound by terms at least as protective of TIBCO as those in this End User License Agreement ("Authorized Users"); (c) sublicense, transfer, assign, distribute to any third party, pledge, lease, rent, or commercially share the Software or any of Customer's rights under this Agreement (for the purposes of the foregoing a change in control of Customer is deemed to be an assignment); (d) use the Software for purposes of providing a service bureau, including, without limitation, providing third-party hosting, or third-party application integration or application service provider-type services, or any similar services; (e) use the Software in connection with ultrahazardous activities, or any activity for which failure of the Software might result in death or serious bodily injury to Customer or a third party; or (f) directly or indirectly, in whole or in part, modify, translate, reverse engineer, decrypt, decompile, disassemble, make error corrections to, create derivative works based on, or otherwise attempt to discover the source code or underlying ideas or algorithms of the Software. Customer may engage in such conduct as is necessary to ensure the interoperability of the Software as required by law, provided that prior to commencing any decompilation or reverse engineering of any Software, Customer agrees to it shall observe strict obligations of confidentiality and provide TIBCO reasonable advance written notice and the opportunity to assist with and/or conduct such activity on Customer's behalf and at Customer's expense; and (g) notwithstanding the method of delivery of the Software, the scope of the licenses granted under this Agreement are limited to the Software as set forth in an Ordering Document.

Beta and Evaluation Licenses. Notwithstanding the foregoing, if the Software is being provided for demonstration, beta testing, or evaluation purposes, then Customer agrees (a) to use the Software solely for such purposes, (b) that the Software will not be used or

deployed in a production or development environment, and (c) that such use shall automatically terminate upon the earlier of thirty days from the date Customer receives the right to install the Software, or Customer's receipt of notice of termination from TIBCO.

Maintenance. Maintenance, if ordered, including first year and all subsequent years, is provided under the policies set forth in the Maintenance Program Guide in effect at the time Maintenance services are provided. The policies set forth in the Maintenance Program Guide, incorporated in this Agreement, are subject to change at TIBCO's discretion; however the level of Maintenance service provided by TIBCO will not be materially reduced during the period for which Maintenance fees have been paid by Customer. The current version of the Maintenance Program Guide can be accessed at <http://www.tibco.com/services/support/default.jsp>. To receive Maintenance, all Software must be properly licensed and annual Maintenance fees paid. TIBCO is not obligated to continue providing Maintenance if annual Maintenance fees have not been paid.

Services. Customer may request additional services ("Services") either in an Ordering Document, or by a separate mutually executed work order, statement of work or other work-request document incorporating the term of this End User License Agreement (each, a "Work Order"). Unless otherwise expressly agreed to in a Work Order, all Services and any work product therefrom shall be (a) performed on a time and materials basis, plus meals, lodging, travel, and other expenses reasonably incurred in connection therewith, (b) deemed accepted upon delivery, and (c) exclusively owned by TIBCO (except for Confidential Information of Customer), including all right, title and intellectual property or other right or interest therein. Each Work Order is intended to constitute an independent and distinct agreement of the parties, notwithstanding that each shall be construed to incorporate all applicable provisions of this End User License Agreement.

Limited Warranty. If Customer obtained the Software directly from TIBCO, then TIBCO warrants that for a period of thirty (30) days from the Purchase Date: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software will substantially conform to its Documentation. This limited warranty extends only to the original Customer hereunder. Customer's sole and exclusive remedy and the entire liability of TIBCO and its licensors under this limited warranty will be, at TIBCO's option, repair, replacement, or refund of the Software and applicable Maintenance fees, in which event this End User License Agreement shall terminate upon refund thereof.

This warranty does not apply to any Software which (a) is licensed for beta, evaluation, testing or demonstration purposes for which TIBCO does not receive a license fee, (b) has been altered or modified, except by TIBCO, (c) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by TIBCO, (d) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (e) is used in violation of any other term of this End User License Agreement. Customer agrees to pay TIBCO for any Maintenance or Services provided by TIBCO related to a breach of the foregoing on a time, materials, travel, lodging and other reasonable expenses basis. If Customer obtained the Software from a TIBCO reseller or distributor, the terms of any warranty shall be as provided by such reseller or distributor, and TIBCO provides Customer no warranty with respect to such Software.

EXCEPT AS SPECIFIED IN THIS LIMITED WARRANTY, THE SOFTWARE, MAINTENANCE AND SERVICES ARE PROVIDED "AS IS", ALL EXPRESS OR IMPLIED CONDITIONS,

REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, SATISFACTORY QUALITY OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW. CERTAIN THIRD PARTY SOFTWARE MAY BE PROVIDED TO CUSTOMER ALONG WITH CERTAIN TIBCO SOFTWARE AS AN ACCOMMODATION TO CUSTOMER. THIS THIRD PARTY SOFTWARE IS PROVIDED "AS IS". CUSTOMER MAY CHOOSE NOT TO USE THIRD PARTY SOFTWARE PROVIDED AS AN ACCOMMODATION BY TIBCO. NO WARRANTY IS MADE REGARDING THE RESULTS OF ANY SOFTWARE, MAINTENANCE OR SERVICES OR THAT THE SOFTWARE WILL OPERATE WITHOUT ERRORS, PROBLEMS OR INTERRUPTIONS, OR THAT ERRORS OR BUGS IN THE SOFTWARE WILL BE CORRECTED, OR THAT THE SOFTWARE'S FUNCTIONALITY, MAINTENANCE OR SERVICES WILL MEET CUSTOMER'S REQUIREMENTS. NO TIBCO DEALER, DISTRIBUTOR, AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS OR ADDITIONS TO THIS WARRANTY.

Indemnity. If Customer obtained the Software from TIBCO directly, then TIBCO agrees at its own expense to defend or, at its option, to settle, any claim or action brought against Customer to the extent it is based on a claim that the unmodified Software infringes any patent issued by the United States, Canada, Australia, Japan, or any member of the European Union, or any copyright, or any trade secret of a third party; and TIBCO will indemnify and hold Customer harmless from and against any damages, costs and fees reasonably incurred (including reasonable attorneys' fees) that are attributable to such claim or action and which are assessed against Customer in a final judgment; provided that TIBCO is promptly notified in writing of such claim, TIBCO has the exclusive right to control such defense and/or settlement, and Customer shall provide reasonable assistance (at TIBCO's expense) in the defense thereof. In no event shall Customer settle any claim, action or proceeding without TIBCO's prior written approval. In the event of any such claim, litigation or threat thereof, TIBCO, at its sole option and expense, shall (a) procure for Customer the right to continue to use the Software or (b) replace or modify the Software with functionally equivalent software. If such settlement or modification is not commercially reasonable (in the reasonable opinion of TIBCO), TIBCO may cancel this End User License Agreement upon sixty days prior written notice to Customer, and refund to Customer the unamortized portion of the license fees paid to TIBCO by Customer based on a five-year straight-line depreciation. This Section states the entire liability of TIBCO with respect to the infringement of any intellectual property rights, and Customer hereby expressly waives any other liabilities or obligations of TIBCO with respect thereto. The foregoing indemnity shall not apply to the extent any infringement could have been avoided by use of the then-current release.

Limitation of Liability. EXCEPT AS PROVIDED UNDER INDEMNITY OR RESULTING FROM A BREACH OF CONFIDENTIALITY (THE "EXCLUDED MATTERS"), IN NO EVENT WILL EITHER PARTY OR TIBCO'S LICENSORS BE LIABLE FOR ANY LOST DATA, LOST REVENUE, LOST PROFITS, DAMAGE TO REPUTATION, BUSINESS INTERRUPTION, OR ANY OTHER INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, PUNITIVE, EXEMPLARY OR ANY SIMILAR TYPE DAMAGES ARISING OUT OF THIS AGREEMENT, THE USE OR THE INABILITY TO USE THE SOFTWARE, OR THE PROVISION OF ANY MAINTENANCE OR SERVICES, EVEN IF A PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT FOR THE EXCLUDED MATTERS, IN NO EVENT SHALL A PARTY BE LIABLE TO THE OTHER, WHETHER IN CONTRACT, TORT (INCLUDING ACTIVE OR

PASSIVE NEGLIGENCE), BREACH OF WARRANTY, CLAIMS BY THIRD PARTIES OR OTHERWISE, EXCEED THE PRICE PAID BY CUSTOMER UNDER THE APPLICABLE ORDERING DOCUMENT.

THE FOREGOING LIMITATIONS SHALL APPLY EVEN IF THE ABOVE-STATED REMEDY OR LIMITED WARRANTY FAILS OF ITS ESSENTIAL PURPOSE. BECAUSE SOME STATES OR JURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO CUSTOMER.

Confidentiality. "Confidential Information" means the terms of this End User License Agreement; all information marked by the disclosing party as proprietary or confidential; any provided software, related documentation or related performance test results derived by Customer; and any methods, concepts or processes utilized in provided software or related documentation. Confidential Information shall remain the sole property of the disclosing party and shall not be disclosed to any non-Authorized User of either TIBCO or Customer without the prior written consent of the disclosing party. If Confidential Information is communicated orally, such communication shall be confirmed as "Confidential" in writing within thirty days of such disclosure. The parties agree to protect the Confidential Information of the other in the same manner it protects the confidentiality of similar information and data of its own (and at all times exercising at least a reasonable degree of care). Except with respect to the Software, items will not be deemed Confidential Information if (i) available to the public other than by a breach of an agreement with TIBCO, (ii) rightfully received from a third party not in breach of any obligation of confidentiality, (iii) independently developed by one party without use of the Confidential Information of the other; (iv) known to the recipient at the time of disclosure (other than under a separate confidentiality obligation); or (v) produced in compliance with applicable law or court order, provided the other party is given reasonable notice of the same. Both parties agree to indemnify the other for any damages the other may sustain resulting from their unauthorized use and/or disclosure of the other's Confidential Information. Such damages shall include reasonable expenses incurred in seeking both legal and equitable remedies. To the extent required by law, at Customer's request, TIBCO shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of TIBCO's applicable fee. Customer agrees to observe obligations of confidentiality with respect to such information.

To the extent TIBCO is exposed to individual personal data owned or otherwise held by Customer during the provision of Services, which is subject to various data protection laws and/or regulations ("Protected Data"), TIBCO agrees to treat such Protected Data in accordance with the Customer Privacy and Security Statement set forth at http://www.tibco.com/customer_privacy_security_statement.jsp (the "Statement"). The policies and procedures set forth in the Statement as well as those set forth in the Data Protection Policy Statement at http://www.tibco.com/resources/data_protection_statement.pdf are in place to meet TIBCO's obligations for the protection, integrity and confidentiality of any Protected Data which exceed TIBCO's standard obligations to safeguard Confidential Information.

Export. Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Customer agrees to comply strictly with all such regulations and agrees to obtain all necessary licenses to export, re-export, or import Software.

Government Use. If the Software, Maintenance or Services are being or have been acquired with U.S. Federal Government funds, or

Customer is an agency, department, or other entity of the United States Government ("Government"), the use, duplication, reproduction, release, modification, disclosure or transfer of the Software, Maintenance or Services, or any related documentation of any kind, including technical data, or manuals, is restricted in accordance with Federal Acquisition Regulation 12.212 for civilian agencies and Defense Federal Acquisition Regulation Supplement 227.7202 for military agencies. The Software, Maintenance and Services are COMMERCIAL ITEMS AS DEFINED BY THE FEDERAL ACQUISITION REGULATION. Use of the Software, Maintenance and Services by the Government is further restricted according to the terms of this Agreement and any amendment hereto.

Orders. An Ordering Document means any purchase order, similar document or agreement requesting Software, Maintenance or Services, which shall be deemed accepted only by issuance of a TIBCO invoice and solely for purposes of administrative convenience. None of the terms of the Ordering Document (other than the Software product name, Number of Units, level of Maintenance, description of Services, and fees due in connection therewith) shall apply for any reason or purpose whatsoever, regardless of any statement on any Ordering Document to the contrary, unless countersigned by an officer of TIBCO. This Agreement constitutes the entire agreement between the parties with respect to the use of the Software, Maintenance and Services, and supersedes all proposals, oral or written, and all other representations, statements, negotiations and undertakings relating to the subject matter hereof. All orders of Software, Maintenance or Services by Customer to TIBCO shall be deemed to occur with or without reference to, under the terms of this End User License Agreement, unless expressly superseded by a signed written agreement between the parties. Neither the license to use the Software granted in this Agreement nor the obligation to pay the license fees set forth above are dependent upon the performance by any party of any Services or the supply of any other software program or product. Software shall be delivered electronically, and delivery deemed complete when the Software is made available for download by Customer.

Termination. Maintenance or Services may be terminated: (a) by either party upon a default of the other, such default remaining uncured for fifteen days from written notice from the non-defaulting party; (b) upon the filing for bankruptcy or insolvency of the other party; (c) by either party upon prior written notice at least sixty (60) days prior to the end of any annual Maintenance term; or (d) by Customer (for Services), upon ten days prior written notice or 30 days prior written notice by TIBCO. Termination of Maintenance or Services shall not terminate this End User License Agreement. Customer may terminate this End User License Agreement in its entirety at any time by destroying all copies of the Software. Upon termination or expiration of this End User License Agreement in its entirety, for any reason, Customer must cease using and return or destroy all copies of the Software. Customer's obligation to pay accrued charges and any fees due as of the date of termination, as well as the sections entitled "Confidentiality", "Limited Warranty" and "Limitation of Liability" shall survive any such termination.

Authority. You hereby represent and warrant that you have full power and authority to accept the terms of this End User License Agreement on behalf of Customer, and that Customer agrees to be bound by this End User License Agreement.

Definitions. In connection with this End User License Agreement Software product definitions are as set forth at <http://tibco.com/software/productdefinitions.jsp> which are hereby incorporated into the End User License Agreement and the following capitalized terms shall have the following meaning: "Customer" means the original purchaser or licensee of the Software and any permitted

successors and assigns; "Documentation" means text material that accompanies the Software on delivery, which describes how to make use of that software. "Enterprise" means an unlimited Number of Units of the Software set forth in an Ordering Document as Enterprise, to be deployed by Customer during a period of one year (or such other period of time set forth in the Ordering Document) from the Purchase Date (the "Enterprise Term"), at which time, the Number of Units then deployed in Production and Non-Production use by Customer becomes fixed and Customer may not thereafter deploy additional Units. During the Enterprise Term, Customer's right to deploy an unlimited Number of Units does not extend to any entity which acquires, is acquired by, merged into, or otherwise combined with Customer. Customer hereby agrees to provide TIBCO, within sixty (60) days after the end of the Enterprise Term, with written notice of the Number of Units deployed at the end of the Enterprise Term by Unit and License Type. In the event Customer elects to renew Maintenance (subject to the section entitled "Termination"), then during the Enterprise Term and for the first annual renewal thereafter, TIBCO agrees the annual Maintenance fee for the Software licensed on an Enterprise basis shall not increase by more than the percentage rate change in the United States Department of Labor - Consumer Price Index for All Urban Consumers (CPI-U) for the twelve month period immediately preceding the anniversary date of Maintenance. "Purchase Date" means the date the Ordering Document is accepted by TIBCO. "Project" means an unlimited Number of Units of the Software set forth in an Ordering Document as Project, to be deployed by Customer solely in connection with the description and scope of the Customer project set forth in the Ordering Document, during a period of one year (or such other time period set forth in the Ordering Document) from the Purchase Date (the "Project Term"), at which time, the Number of Units then deployed in Production and Non-Production use by Customer becomes fixed and Customer may not thereafter deploy additional Units. During the Project Term, Customer's right to deploy an unlimited Number of Units does not extend beyond the scope of the Project as set forth herein or to any entity which acquires, is acquired by, merged into, or otherwise combined with Customer. Customer hereby agrees to provide TIBCO, within sixty (60) days after the end of the Project Term, with written notice of the Number of Units deployed at the end of the Project Term by Unit and License Type. In the event Customer elects to renew Maintenance (subject to the section entitled "Termination"), then during the Project Term and for the first annual renewal thereafter, TIBCO agrees the annual Maintenance fee for the Software licensed on a Project basis shall not increase by more than the percentage rate change in the United States Department of Labor - Consumer Price Index for All Urban Consumers (CPI-U) for the twelve month period immediately preceding the anniversary date of Maintenance. "Severity" means the impact levels of an error, defect or malfunction, which under Maintenance are determined by TIBCO as follows: "Severity 1" is an emergency production situation where the Software is totally inoperable or fails catastrophically and there is no workaround; "Severity 2" is a detrimental situation (and there is no workaround) where (a) performance degrades substantially under reasonable loads causing a severe impact on use, (b) the Software is usable but materially incomplete; or (c) one or more mainline functions or commands is inoperable; "Severity 3" is where the Software is usable, but does not provide a function in the most convenient manner; and "Severity 4" is a minor problem or documentation error. "Software" means the most current, generally available object code version on all Platforms then currently available. Software does not include multiple Platforms if the software product licensed on a Platform specific basis as designated in the Software product name or as otherwise listed in an Ordering Document, including its Documentation and any subsequent Updates provided under Maintenance. For the purposes of the product definitions where "TIBCO Software" or "Licensor Software" is used it shall have the same meaning as "Software". "TIBCO" means TIBCO Software Inc.

and any entities, regardless of corporate status, controlled by, controlling, or under common control with TIBCO Software Inc. "Updates" means Software bug fixes, enhancements, and updates, if and when made generally available by TIBCO. Updates provided under Maintenance, warranty or which are provided for any other reason by TIBCO, or TIBCO's authorized resellers or distributors (if applicable), are subject to the license rights, limitations and restrictions of the End User License Agreement.

Special Product Provision. Embedded/Bundled Products. Some Software embeds or bundles other Software. Use of such embedded or bundled Software is solely to enable the functionality of the Software licensed under this Agreement, and may not be used or accessed by any other Software, or for any other purpose. **Bundle.** A collection of Software listed in TIBCO's then current price book to be sold together under a collective name such as "XXX Bundle" which consists of X, Y and Z. The Software which comprises a Bundle must be used in accordance with any specific license restrictions imposed in this Agreement and solely in conjunction with the components of the Bundle; provided that a Bundle component may be accessed by or communicate with other Software separately licensed by Customer. In no event may the Software which comprises a Bundle be used on a standalone basis. A Bundle is sold at a discount to the cost of licensing the individual components due to the restrictions imposed on the use of the Bundle by this section and any specific license restrictions imposed by this Agreement. If the terms of this Agreement with regard to a Bundle are breached, and such breach is not cured within ten (10) days of TIBCO notifying Customer in writing of the breach, TIBCO shall be entitled to invoice Customer for the list price of the individual components of the Bundle based upon the manner in which Customer has deployed and is making use of the components. **Open Source Software.** If Customer uses open source software or any other third party software not supplied by TIBCO in conjunction with the Software, Customer must ensure that its use does not: (i) create, or purport to create, obligations of use with respect to the Software; or (ii) grant, or purport to grant, to any third party any rights to or immunities under TIBCO's intellectual property or proprietary rights in the Software. Customer also may not combine Software with programs licensed under the GNU General Public License ("GPL") in any manner that could cause, or could be interpreted or asserted to cause, the Software or any modifications thereto to become subject to the terms of the GPL. **Third Party Software.** Third Party Software means third-party software identified by its company and/or product name, the provision of which by TIBCO is made solely as an accommodation and in lieu of Customer purchasing a license to Third Party Software directly from the third party vendor.

General. All payments of fees due shall be made in U.S. dollars, net 30 from Purchase Date, or, for any other amounts coming due hereafter, net 30 from TIBCO's invoice. Fees do not include sales, use, withholding, value-added or similar taxes, and Customer agrees to pay all sales, use, value-added, goods and services, consumption, withholding, excise and any other similar taxes or government charges, exclusive of TIBCO's income tax. Customer agree to pay all reasonable costs incurred (including reasonable attorneys' fees) in collecting past due amounts. Except as set forth in the sections entitled "Limited Warranty" and "Indemnity" all fees paid under or in connection with this End User License Agreement are non-refundable and no right of set-off exists. A service charge of one and one-half percent per month will be applied to all invoices that are not paid on time. No delay in the performance of any obligation by either party, excepting all obligations to make payment, shall constitute a breach of this End User License Agreement to the extent caused by force majeure. Customer hereby grants TIBCO and its independent auditors the right to audit Customer's compliance with this End User License Agreement. If any portion of this End User License Agreement is

found to be void or unenforceable, the remaining provisions shall remain in full force and effect.

Governing Law. This End User License Agreement shall be governed by and construed in accordance with the laws of the State of California, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. The United Nations Convention on Contracts for the International Sale of Goods is excluded from application hereto.

Version 5.6, April 2008

Index

A

activities

Parse Copybook Data [29](#)

Render Copybook Data [34](#)

C

Copybook schema shared configuration [20](#)

customer support [viii](#)

P

Parse Copybook Data activity [29](#)

R

Render Copybook Data activity [34](#)

S

shared configuration

Copybook schema [20](#)

support, contacting [viii](#)

T

technical support [viii](#)