

ChainBuilder ESB

Visual Enterprise Integration™

Version 1.1

X12/HL7 Editor Guide



©Copyright 2007
Bostech Corporation
2800 Corporate Exchange Drive
Suite 260
Columbus, OH 43231

Acknowledgements

This document contains proprietary information that is the property of Bostech Corporation. Any reproduction, disclosure, or transfer of this document or the information contained herein without the express written consent of Bostech Corporation is strictly prohibited.

The use of the information contained in this document and the implementation of any of its techniques are the sole responsibility of the client and depend on the client's ability to evaluate the information and implement it into the client's operational environment.

Except for any express written warranties made by it, Bostech Corporation makes no warranties or representations with respect to any information contained herein, whether express, implied, statutory, or otherwise, in fact or in law, including without limitation, any implied warranties of merchantability or fitness for a particular purpose; and in no event shall Bostech Corporation be liable for any special, consequential, indirect, punitive, or exemplary damages in connection with the use of the information contained herein. The information contained in this document is subject to change at any time without notice.

Trademarks

The following trademarks and acknowledgments apply to the information presented in this manual:

- ChainBuilder is a registered trademark of Bostech Corporation.
- Adobe and Acrobat Reader are registered trademarks of Adobe, Inc.
- Java is a registered trademark of Sun Microsystems, Inc.
- Windows (NT, 2000, XP, and Server 2003), .NET Framework, Internet Information Services (IIS) are registered trademarks of Microsoft Corporation.

Credits

The following third-party products are used within the ChainBuilder product, and acknowledgments apply to the information presented in this manual:

- Acrobat Reader is created and licensed by Adobe, Inc.
- This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)
- This product includes software developed by Eclipse (<http://www.eclipse.org/>)

Table of Contents

1. Introduction.....	1
1.1. Overview	1
1.2. Introduction to X12 and HL7 Messages	1
1.2.1. X12	1
1.2.2. HL7.....	1
1.3. Introduction to Variant Definitions in ChainBuilder ESB	2
1.4. Introduction to the Variant Editors	4
2. Starting the Variant Editor	6
2.1. Creating a New Variant.....	6
2.2. Opening an Existing Variant.....	14
3. Editing a Message Definition.....	14
3.1. Creating a New Message	15
3.2. Editing an Existing Message.....	17
3.3. Segments.....	17
3.4. Loops	18
4. Editing a Segment Definition	20
4.1. Creating a New Segment.....	20
4.2. Editing an Existing Segment	21
4.3. Fields	22
4.4. Composite Fields.....	23
5. Testing a Message Definition.....	25
6. ChainBuilder ESB Community.....	30

1. Introduction

1.1. Overview

This document will detail the process of using the X12 and HL7 editors to create and modify X12 and HL7 data formats in ChainBuilder ESB.

1.2. Introduction to X12 and HL7 Messages

1.2.1. X12

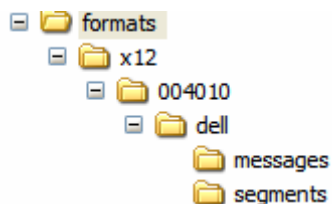
An X12 message is a tilde-separated list of segments, each of which are asterisk-delimited fields. Additionally, loops may be defined which are groups of repeating segments; each loop may have its own rules for the minimum and maximum number of repetitions. X12 definitions in ChainBuilder ESB exist at the transaction level, and do not include outer wrapper information such as the ISA, GS, GE, or IEA segments.

1.2.2. HL7

An HL7 message is a list of segments usually separated by the pipe character (|). The segments may occur in hierarchical loops which indicate the group of segments are optional or repeating as a unit. Each segment is made up of fields, which may also be broken down into sub-fields. The delimiters used in a message can vary from message to message and are defined in the first segment of the message.

1.3. Introduction to Variant Definitions in ChainBuilder ESB

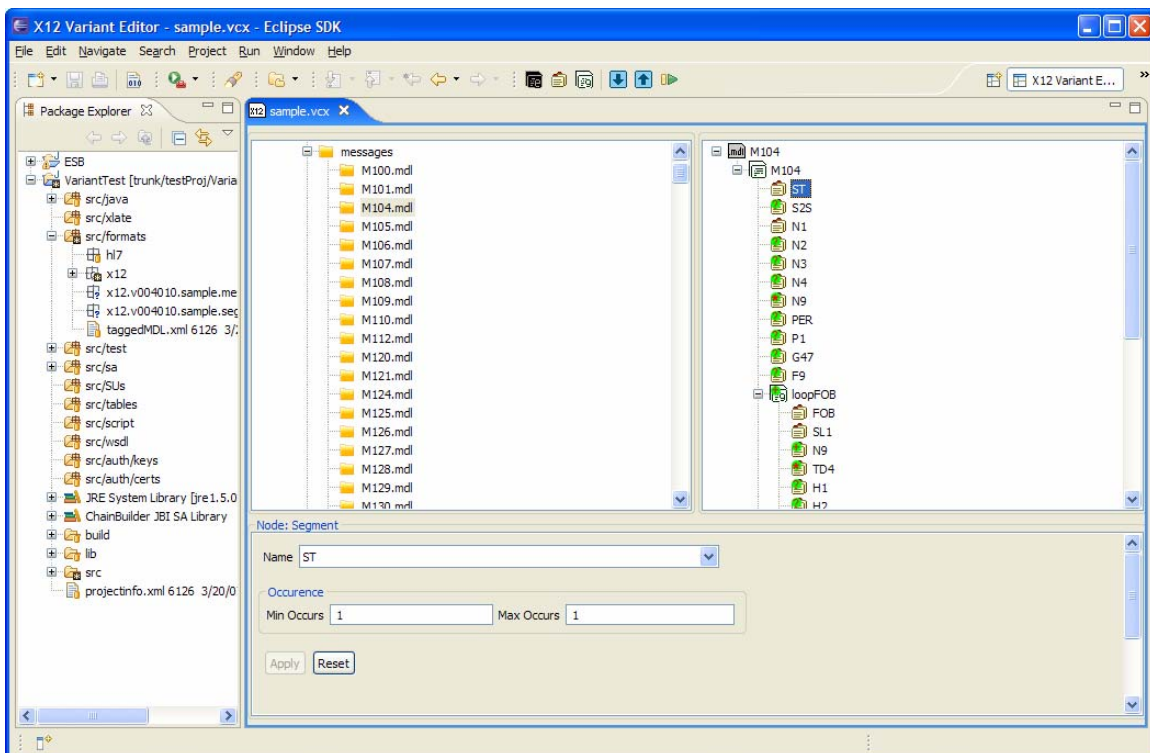
ChainBuilder ESB allows for the creation of new X12 and HL7 message definitions and the modification of existing definitions. All of these changes from the base packages are managed within variants. Message variants can be created and stored at either the ESB Project level or at the Service Assembly Project level. Variants stored in the ESB Project level will be available to all associated Service Assembly projects, while variants created at the Service Assembly level will only be available within that particular Service Assembly. In either case, the variant will be stored under the formats/x12 or formats/hl7 directory in a version-specific folder. For example, an X12 variant named “dell” would be stored in the following directory structure:



1.4. Introduction to the Variant Editors

The X12 and HL7 Editors operate in the same manor, so they will be referred to generically as "Editor" or "Variant Editor". If there is a difference in operation between the two editors, a special note will be made.

The editor is divided into three separate views. The top left view provides a tree structure listing the version and variant currently open, as well as the available messages and segments defined within that particular variant. Additionally, this panel is used to create new messages and segments. The upper right of the editor provides information about the object that is currently selected in the tree view. This panel is used for editing and defining messages and segments. The lower panel contains configuration information for the object which has currently been selected in the upper right area. An example has been provided below.

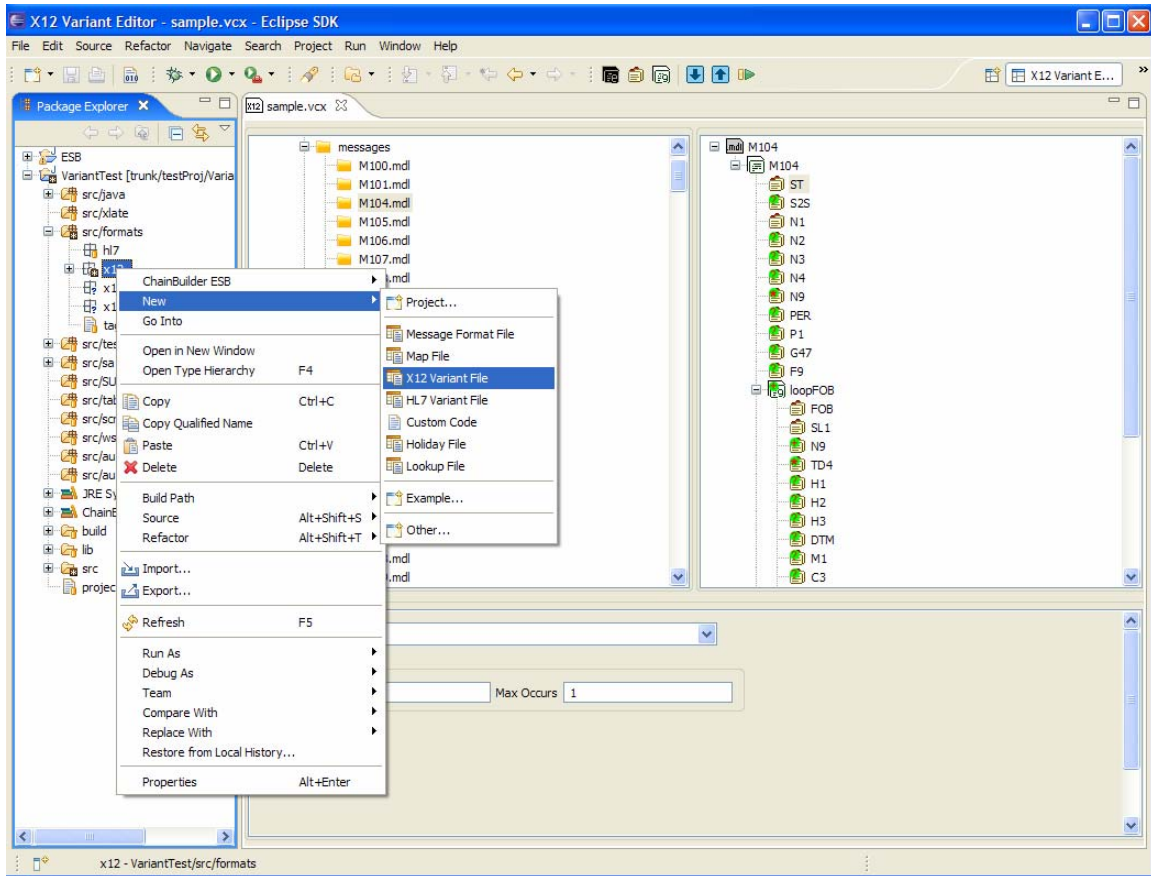


2. Starting the Variant Editor

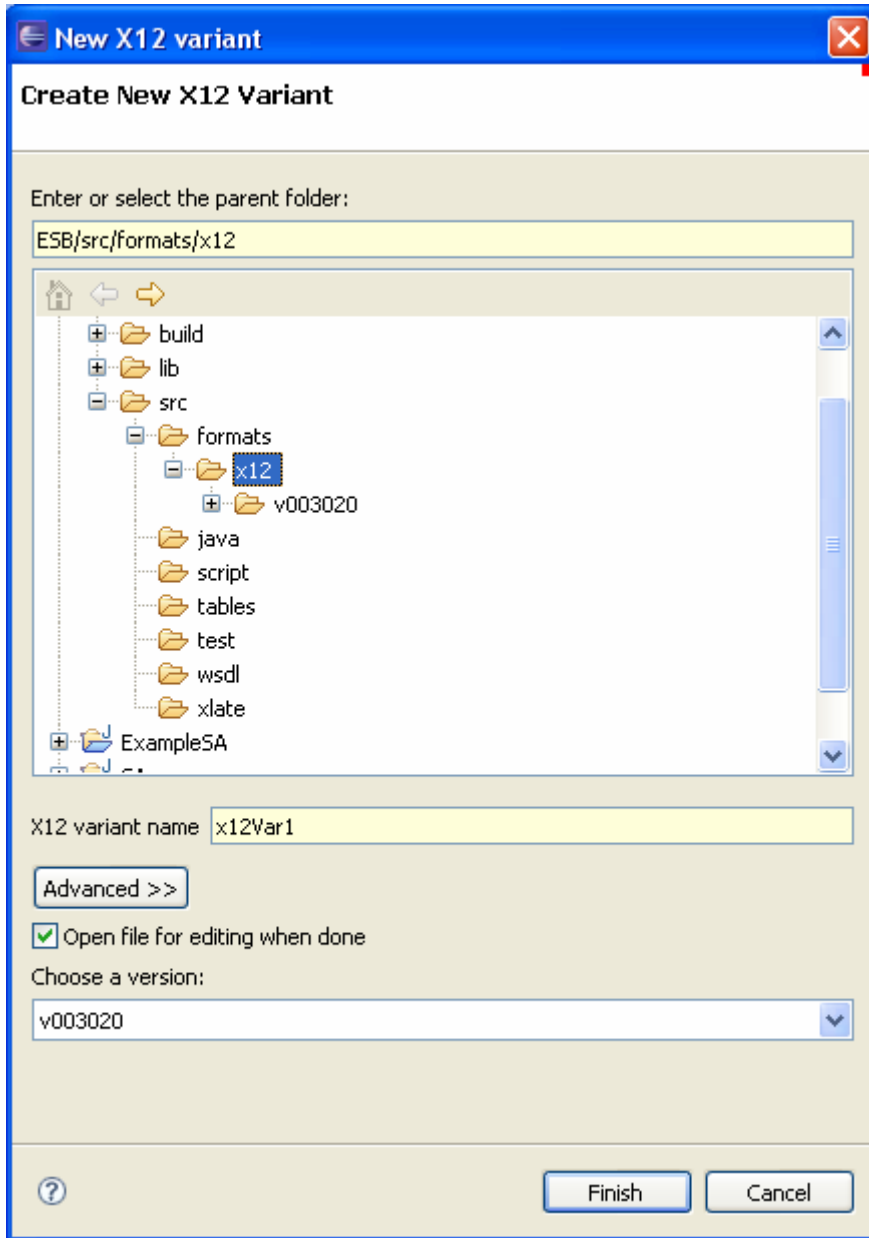
The editor allows for both the creation of new variants and the modification of previously-created variants. It is necessary to first create a variant before modifying any X12 message or segment definitions; the predefined X12 and HL7 structures are not available for modification.

2.1. Creating a New Variant

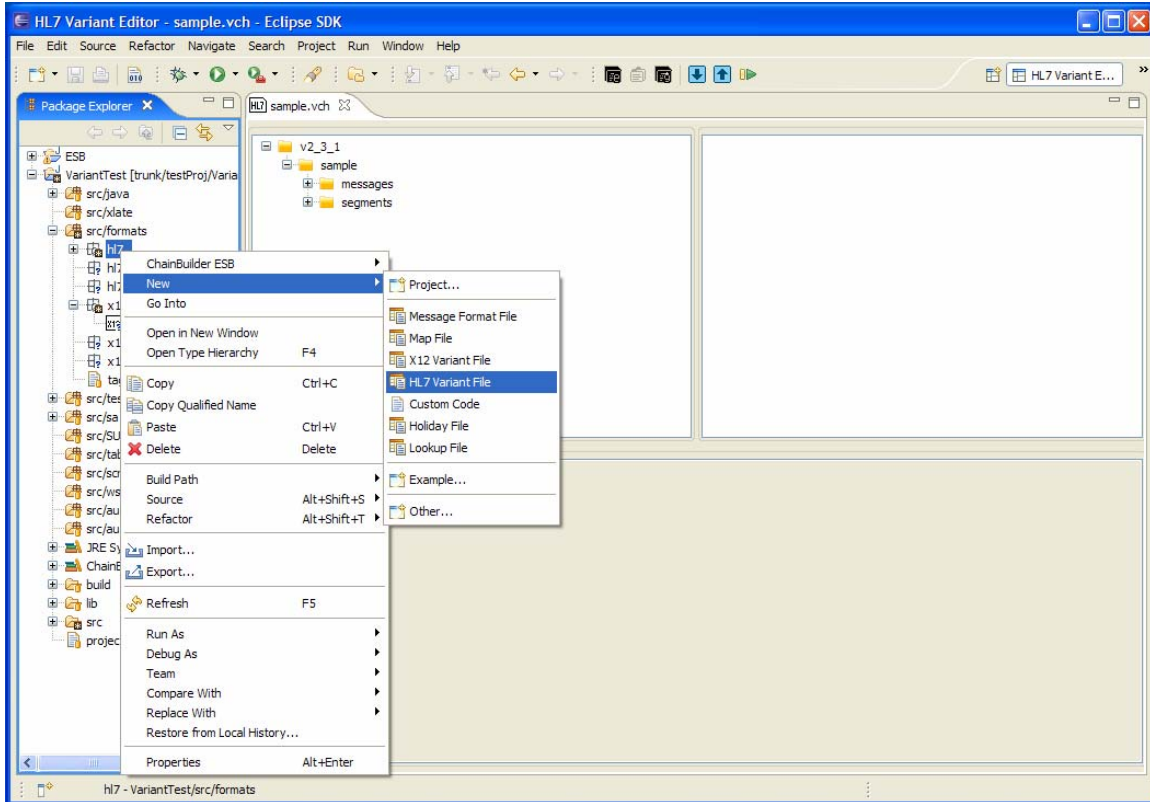
In order to create a new X12 variant, right-click on the `src/formats/x12` directory within the Package Explorer and select `New -> X12 Variant File`.



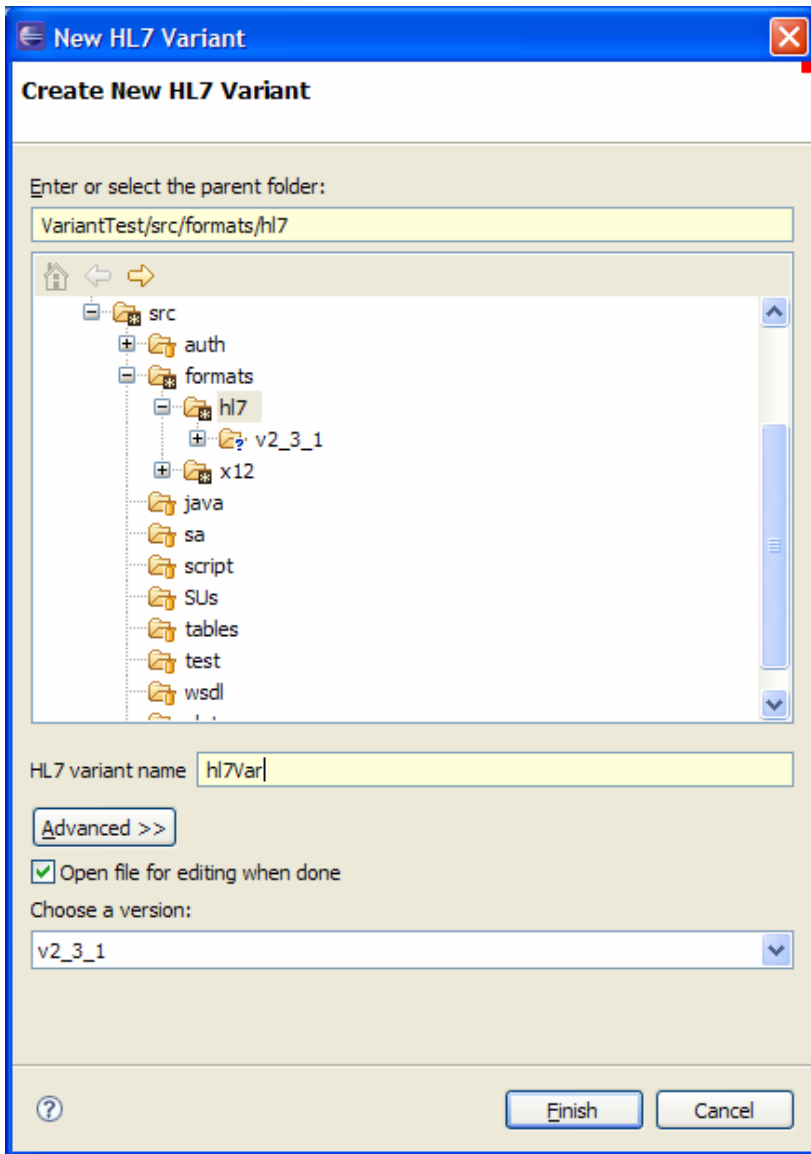
The New X12 Variant screen will now open, allowing the user to select the folder and file name. If desired, modify the X12 variant name to an appropriate file name while retaining the .vcx extension. Next, from the drop-down menu near the bottom of the window select the version of X12 on which the variant should be based. Finally, click Finish to complete the creation of the variant and open the X12 variant editor.



In order to create a new HL7 variant, right-click on the `src/formats/hl7` directory within the Package Explorer and select `New -> HL7 Variant File`.



The New HL7 Variant screen will now open, allowing the user to select the folder and file name. If desired, modify the HL7 variant name to an appropriate file name while retaining the `.vch` extension. Next, from the drop-down menu near the bottom of the window select the version of HL7 on which the variant should be based. Finally, click `Finish` to complete the creation of the variant and open the HL7 variant editor.

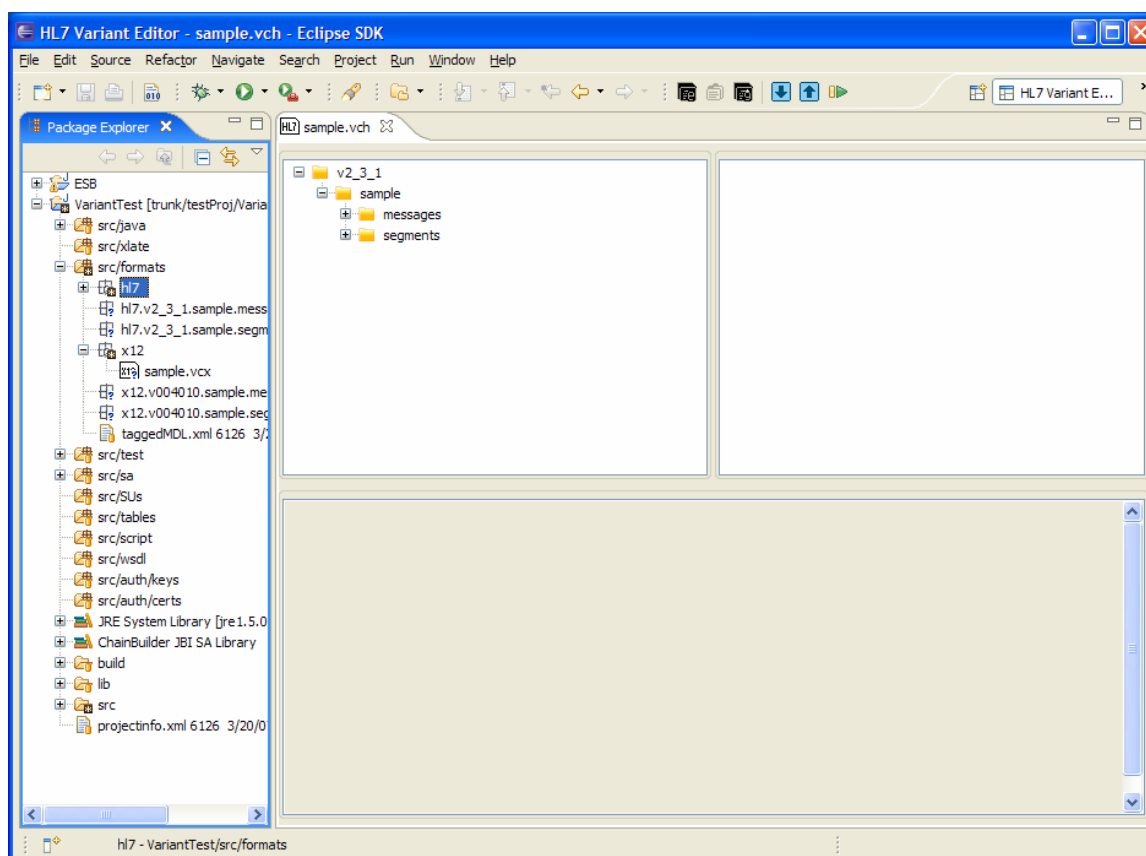


2.2. Opening an Existing Variant

To open an existing variant, double-click on the .vcx or .vch file that was previously created. Variants are stored in the src/formats/x12 and src/formats/hl7 folders in the Package Explorer.

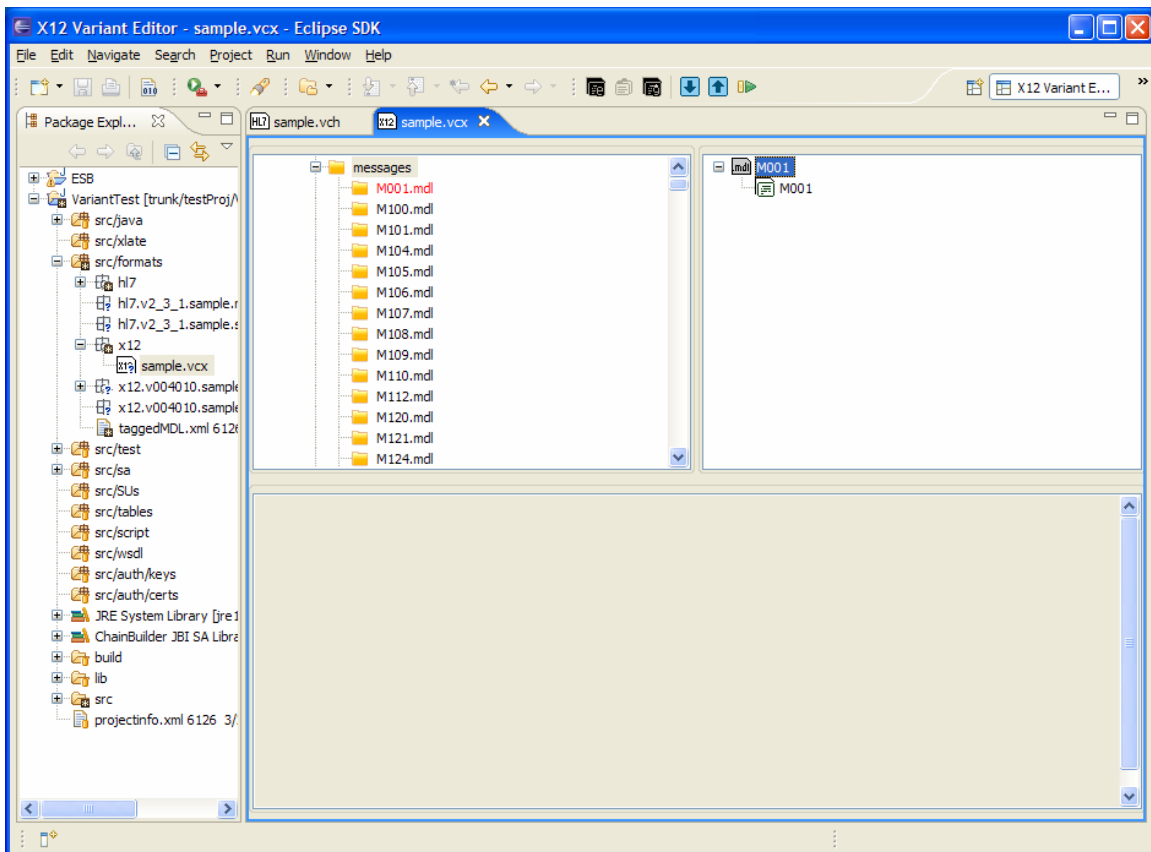
3. Editing a Message Definition

When the variant editor is opened, the view presented will show the version, name of the variant, and the messages and segments folders.



3.1. Creating a New Message

To create a new message, right-click on the messages folder in the editor (not in the Package Explorer) and choose New Message. A dialog box will open asking the user to enter the new message name. Enter the desired message name and click OK. A new message definition will be created and opened in the editor. Initially, the message will contain no segments.



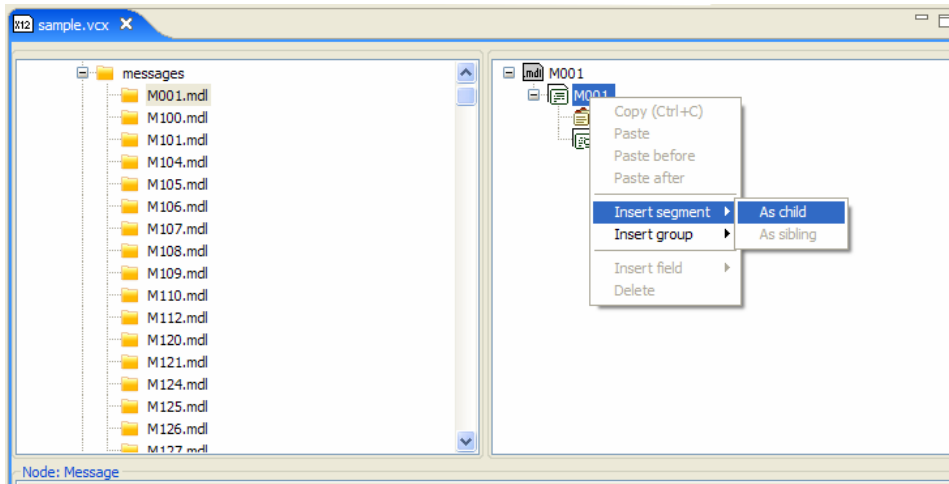
Now that the new message definition has been created, the message can be defined by adding segments to it.

3.2. Editing an Existing Message

After opening the variant editor, open the messages folder and double-click on the desired message. The current message definition will appear on the right side of the editor window.

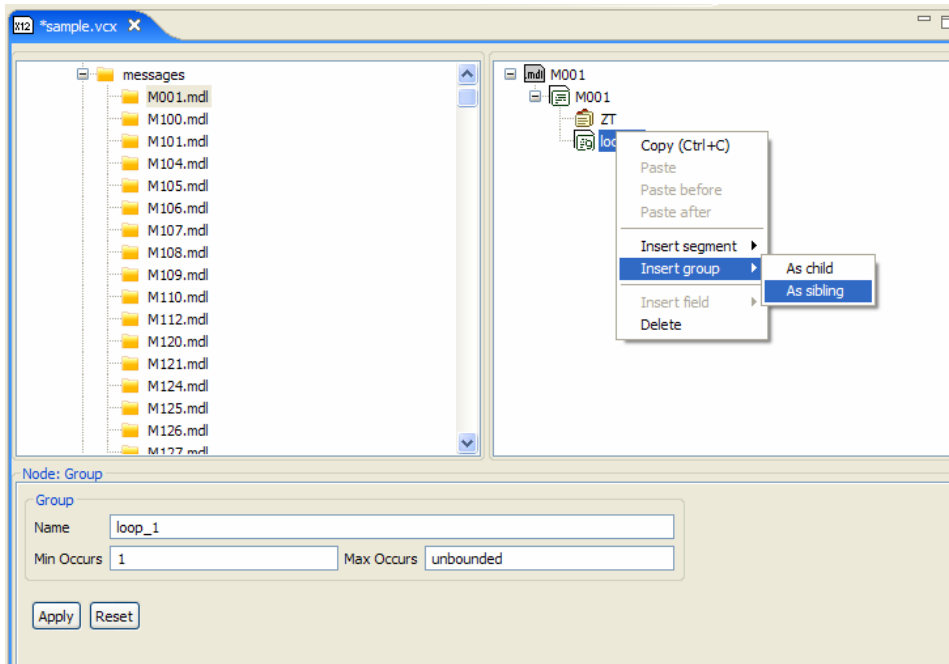
3.3. Segments

Adding a segment to an existing message can be accomplished by right-clicking on the message node and selecting “Insert segment” -> “as child” or by right-clicking on an existing segment within the message definition and selecting “Insert segment” -> “as sibling”.



3.4. Loops

When editing a message format, repeating loops can be easily added by selecting the “Insert group” option from the right-click menu either as a Child when selected on the message name or as a sibling when selected on another segment within the message. Groups can also be added as children of other groups to create a nested repeating definition. When the group element is selected, the lower panel contains configuration options. The Min Occurs and Max Occurs values set the minimum and maximum number of times that the loop may occur within the message. Both fields can be filled with numbers, additionally the Max Occurs field may contain “unbounded” to indicate that an infinite number of repetitions are permitted.

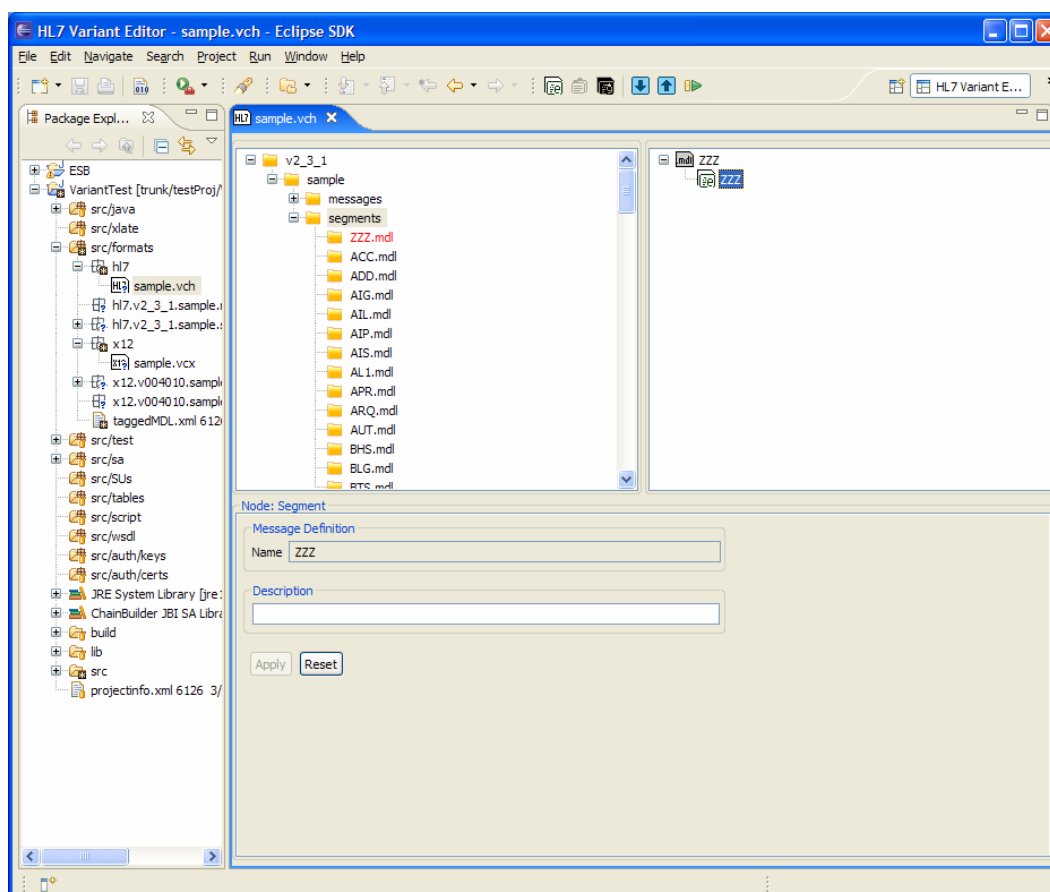


4. Editing a Segment Definition

The following sections contain information about creating new segments and modifying existing segments within the variant editor.

4.1. Creating a New Segment

Within the editor, it is possible to modify segment definitions and to create new segment definitions. The creation of new segments works similarly to the creation of new message definitions; the user should right-click on the Segments folder in the editor and select “New Segment”. This will open the New Segment dialog. Enter the name of the new segment and click Finish.

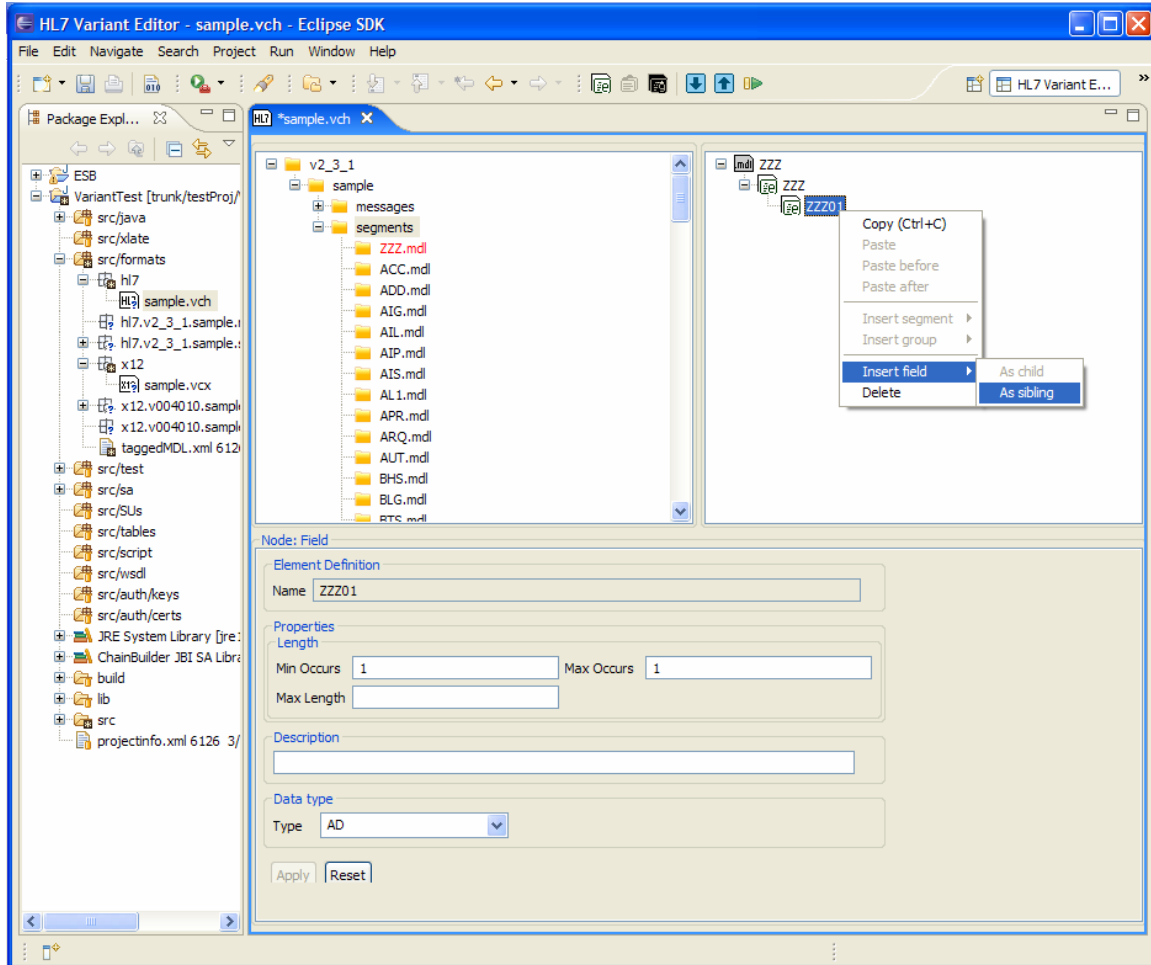


4.2. Editing an Existing Segment

Double-clicking on the name of a segment inside the segments folder within the editor will open the right panel for the purpose of editing the selected segment. Select an existing field, then right-click it to see the list of available actions. Selecting Delete will remove the selected field. The Insert Field options may be used to insert additional fields.

4.3. Fields

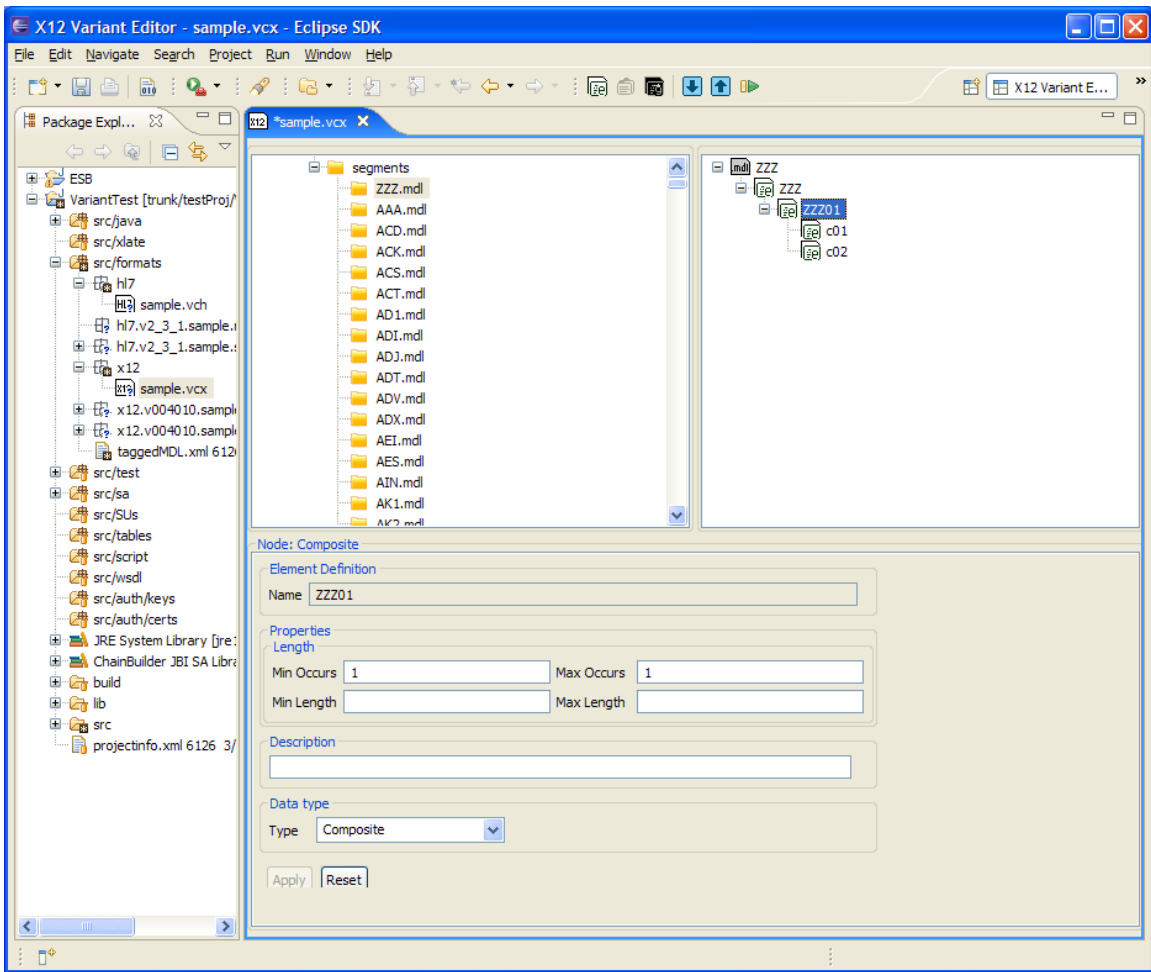
Fields are the lowest-level containers for holding individual data elements. Fields may be added to a segment by right-clicking in the segment definition and selecting the Insert Field option. The name is automatically generated based on its position in the segment.



4.4. Composite Fields

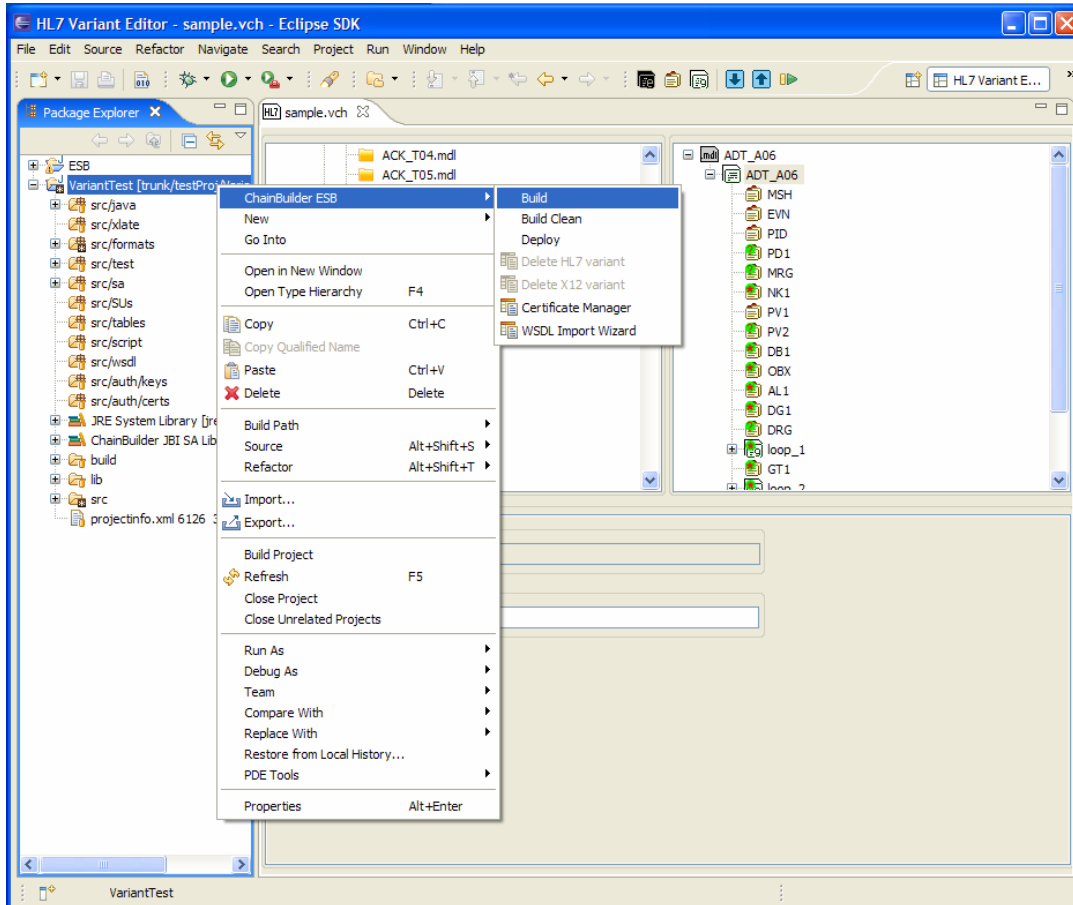
Composite fields are a special type of field available in X12 only. A Composite field allows a field to be broken down into sub-fields. To define a composite, set the Type setting of a field to "Composite". After doing this, you may add sub-fields to it by right-clicking on the field and selecting the "Insert Field" -> "as child" option. This option is only available if the field is defined as a composite.

The screenshot below shows a field ZZZ01 that is defined as a composite and contains 2 subfields "c01" and "c02".

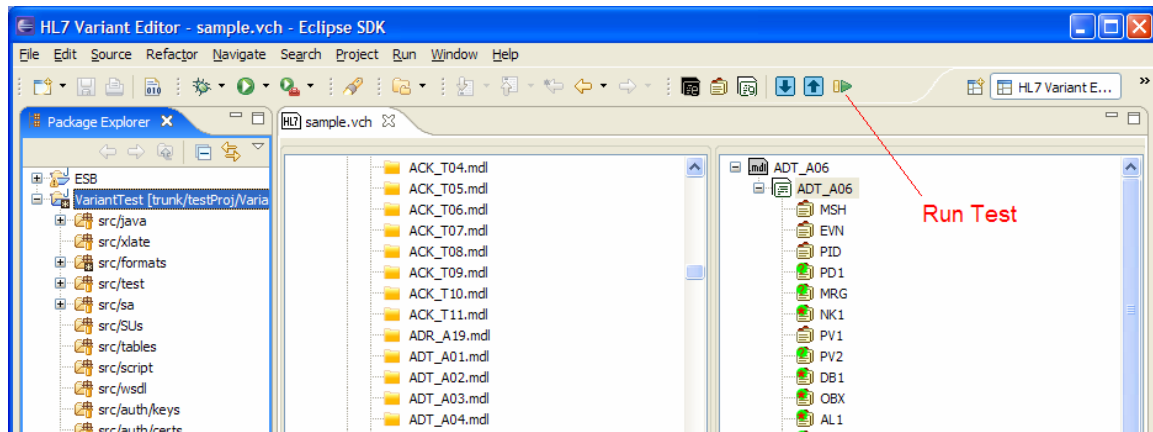


5. Testing a Message Definition

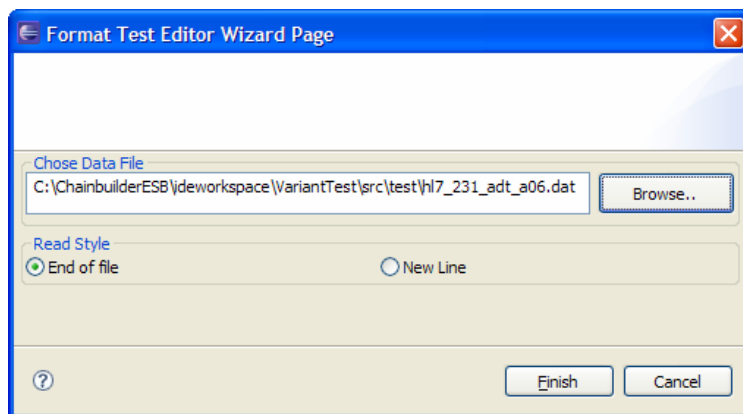
The Variant Editors allow you to test a message definition using sample data to verify that the message parses as expected. Before you can test your definition, you must build your project. This is done by right-clicking on the project in the Package Explorer and selecting "ChainBuilder ESB" -> "Build".



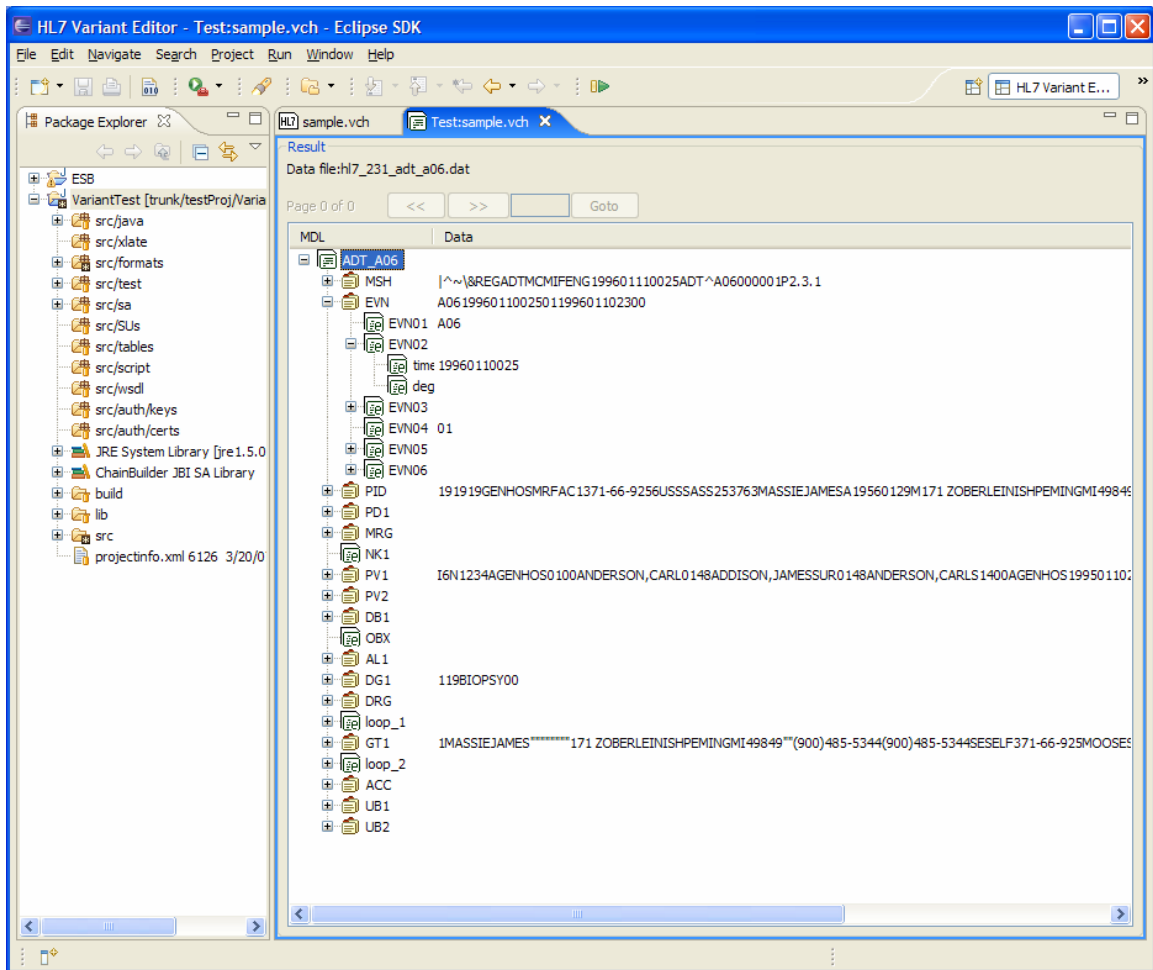
Once the project has been built successfully, select the message you would like to test so the definition is displayed in the panel on the right. Then click on the Run Test button in the toolbar, it the button indicated in the screenshot below.



Then select a data file that contains a sample message that matches the selected message definition. If the Read Style is set to "End of file", then the entire contents of the sample file will be used as a single message. If "New Line" is selected, then each line in the file is considered an individual message. Click Finish to run the test.



The parsed data is then displayed in a tree view. Each segment can be expanded to show the data contained in the individual fields and subfields.



6. ChainBuilder ESB Community

ChainForge.net is the internet's premier destination to share ChainBuilder and JBI knowledge with your peers.

Join the ChainBuilder ESB Community:

<http://www.chainforge.net/community>

As a member you can view content and contribute to a Forum:

<http://www.chainforge.net/community/forums.html>

Read ChainBuilder ESB related Blogs:

<http://www.chainforge.net/blogs>