

# ABR User's Guide

## Dynalab Test Systems



# Table of Contents

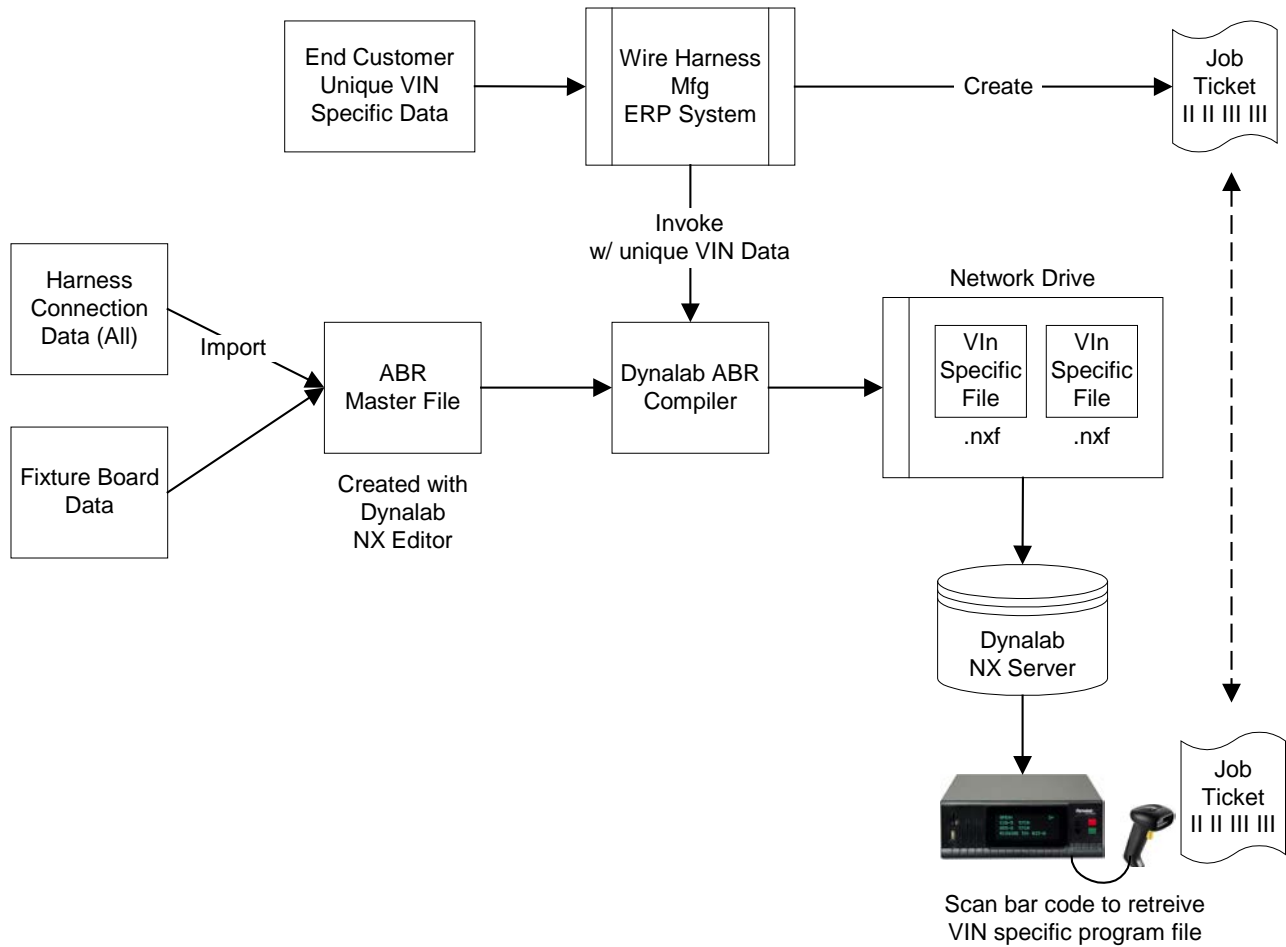
TABLE OF CONTENTS.....	2
ABR OVERVIEW.....	3
CREATING AN ABR MASTER PROGRAM FILE.....	5
COMPILING AN ABR PROGRAM.....	8
ABR PROGRAM EXAMPLE.....	9
<i>Testing a Wire Harness with No Post-Assembled Parts</i> .....	9
<i>Testing a Wire Harness with Post-Assembled Parts</i> .....	22
PROGRESSIVE ASSEMBLY EXAMPLE.....	26

© Copyright, 2020, Dynalab Test Systems, Inc. All rights reserved.

All features/functions mentioned within are subject to change. This document is for informational purposes only. Dynalab Test Systems, Inc., makes no warranties, expressed or implied, in this document. Dynalab® and NX® are trademarks of Dynalab Test Systems, Inc.

# ABR Overview

The below diagram illustrates how the NX Editor ABR Master File, ABR Compiler and NX Server work together with VIN specific data to generate and execute unique program files.

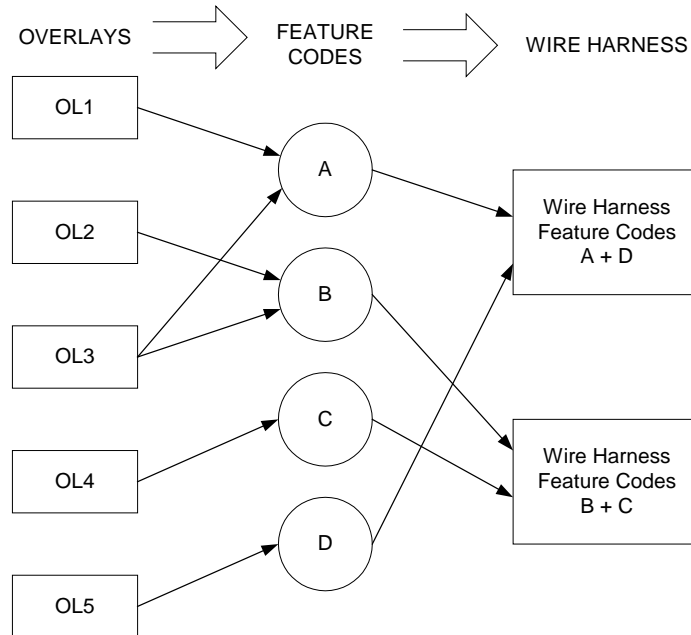


The NX Editor is optionally available with functionality designed to support feature-based program compilation. This functionality is designated ABR. A special license is required to activate the ABR features.

In certain wire harness manufacturing environments, each wire harness is a unique entity comprised of a collection of pre-defined sub-harnesses. The collection of sub-harnesses that make up the wire harness represents a set of customer-required features. This scenario is common for wire harnesses that are manufactured for installation in low-volume, highly-optionable products such as large trucks or buses where each truck or bus represents a unique combination of features that have been ordered by the customer, and wire harnesses that support those features must be fabricated for each truck or bus.

The NX Editor ABR feature is capable of compiling a program for every wire harness to be tested where each wire harness is a potentially unique combination of a pre-defined set of sub-harnesses (or overlays). Each feature that a customer can order is comprised of one or more overlays.

These logical relationships are depicted below. In this example, feature code A is made of overlays OL1 and OL3, feature code B is made up of overlays OL2 and OL3, feature code C is made up of overlay OL4 and feature code D is made up of overlay OL5. Finally, the example shows two wire harnesses: one comprised of feature codes A and D, the other comprised of feature codes B and C.



The NX Editor's ABR feature provides the capability of defining each overlay and defining the mappings of overlays to feature codes. Once these are defined, the ABR compiler enables the user to generate a test program for a particular wire harness that is defined as a set of feature codes.

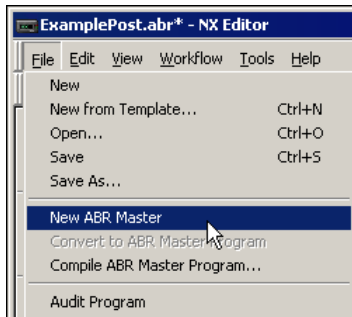
# NX Editor - ABR Enabled

The current release of the standard NX Editor software application includes ABR functions. However, these functions are only enabled when the NX Editor is installed with a specific license number. Contact Dynalab to purchase and obtain a license number. Part Number 5-5008.

## Creating an ABR Master Program File

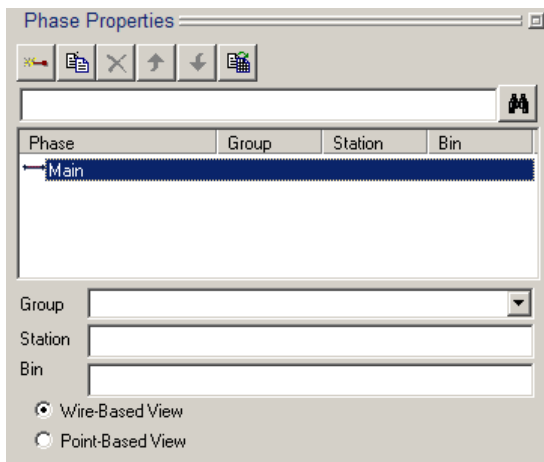
An ABR Master program file must first be created. This file defines all overlays, feature codes, mappings of overlays to feature codes, and the workflow. The ABR Master program then is compiled for a selected set of feature codes to create a unique test program for a wire harness.

In order to create a new ABR Master program, select “New ABR Master” from the File menu:



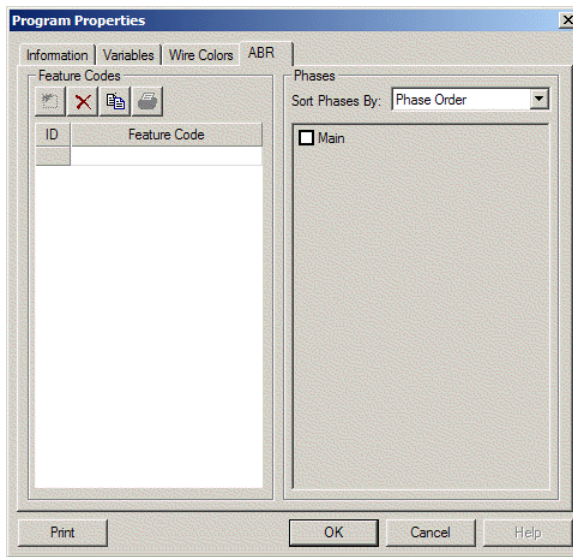
This makes the following features available:

### **Expanded Phase Properties**



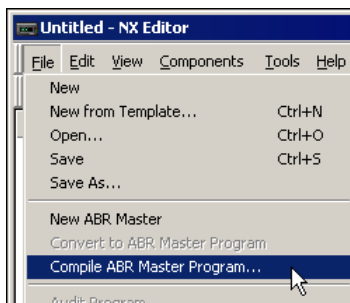
The Phase Properties pane now has entries for associating a Group, Station, and Bin with each Phase. These new attributes are explained in the subsequent programming examples in this document.

### ABR Feature Code to Phase association window



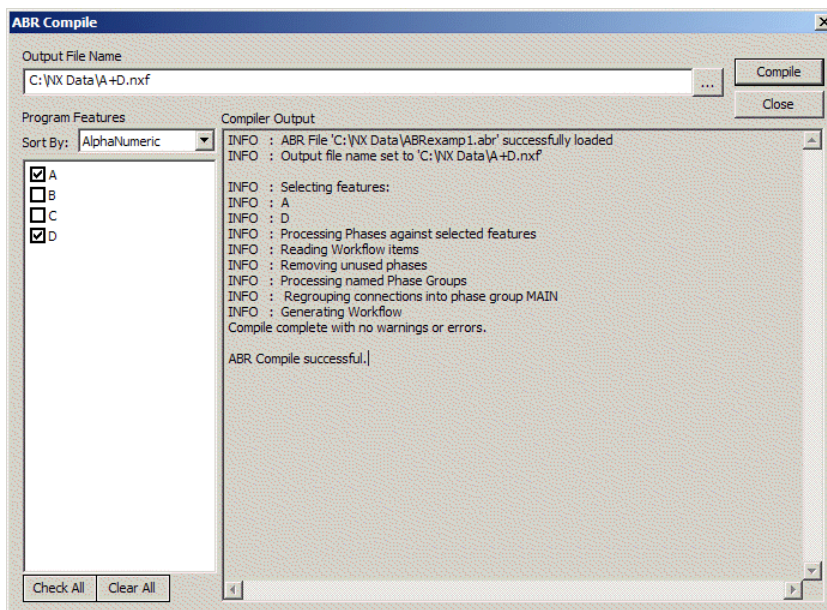
The Program Properties window now has a new tab labeled “ABR”. This tab allows for the association of Feature Codes to Phases (or overlays). A more detailed explanation is provided in the subsequent programming examples in this document.

### Compile ABR Master Program Selection in File Menu



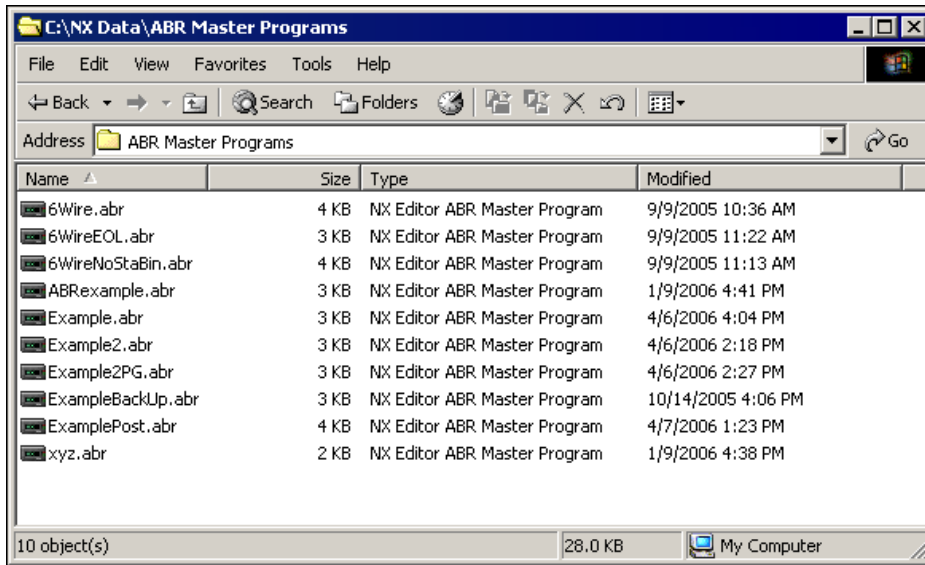
The File pull down menu has an entry labeled “Compile ABR Master Program”. This is used to compile a NX program for a specific set of Feature Codes. A more detailed explanation is provided in the subsequent programming examples in this

### Compile ABR Program Window

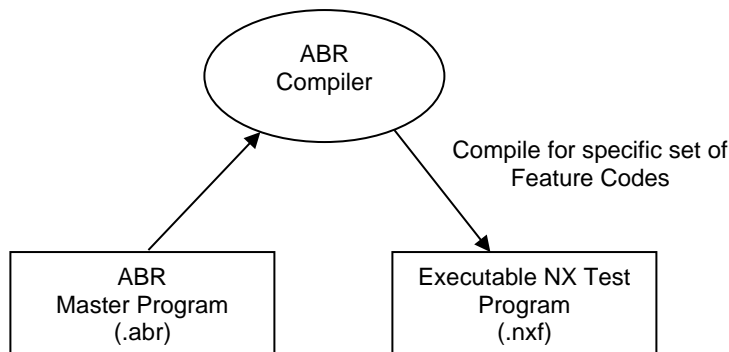


The ABR Compile window displays the results of the ABR Compiler.

Master ABR files are saved with a .ABR extension. They can only be opened by ABR-enabled versions of the NX Editor. These files cannot be executed on a NX Tester.



Master ABR files are used to compile executable test programs for a wire harness defined by a specific set of Feature Codes:



# **Compiling an ABR Program**

An ABR program can be compiled in one of three ways:

- 1 Using the “Compile ABR Master Program” menu choice within the ABR-enabled NX Editor, as shown above.
- 2 Using the spreadsheet entitled **DynalabABRCompiler.xls**. Refer to the spreadsheet’s Instructions tab for details on how to use it.
- 3 Executing the compiler from another program. This is documented in the Application Note entitled *NX ABR Compiler-Calling Interface*.

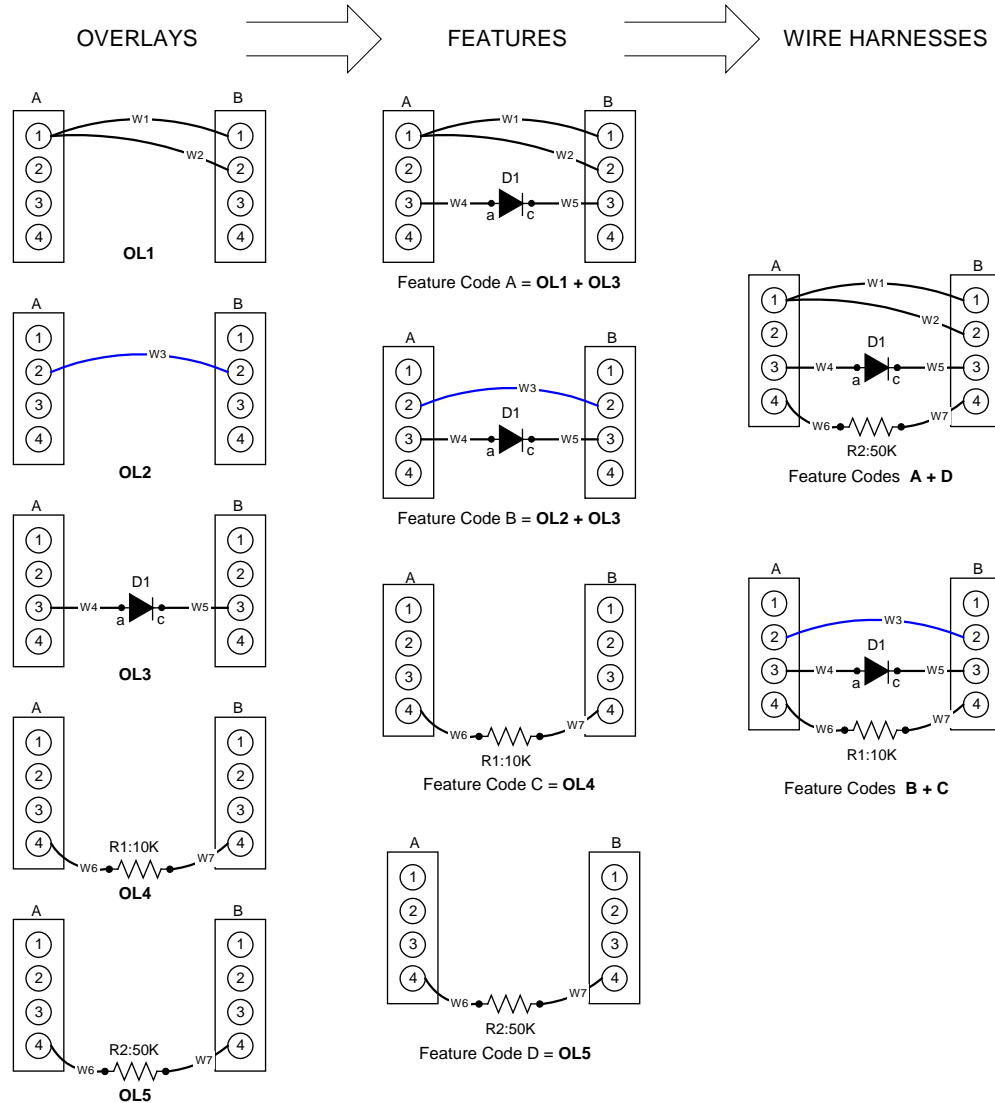
The examples provided in the subsequent sections of this document provide further explanation of these ABR functions and features.



# ABR Program Example

## Testing a Wire Harness with No Post-Assembled Parts

The easiest way to explain how to use the ABR functionality of the NX Editor is by way of a simple example. Consider the following relationships between overlays, feature codes, and final wire harnesses:

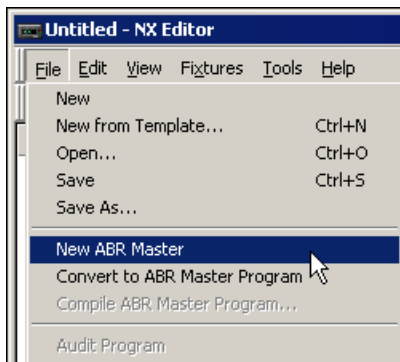


The following is a list of the tasks required to implement the example, using the NX Editor with the ABR feature enabled:

- 1 Create a new ABR master file
- 2 Define the fixture blocks in the Fixture Blocks view.
- 3 Define all components (one diode and two resistors) in the Components view.
- 4 In the Connections View, define a phase for each overlay, associate each phase with a Phase Group named MAIN, and define all connections for each phase.
- 5 Define each Feature Code and define the associated overlays for each Feature Code – this is done in the Program Properties -> ABR window
- 6 Modify the Workflow
- 7 Audit the program
- 8 Compile a wire harness test program for a selected set of Feature Codes

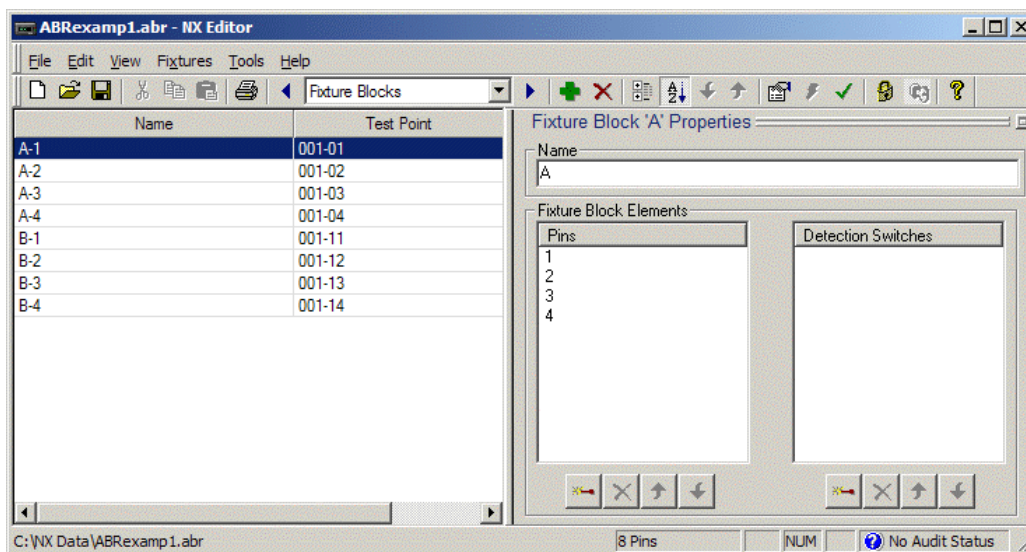
#### Step 1: Create new ABR master file

From the File menu, select “New ABR Master”. This enables the ABR features.



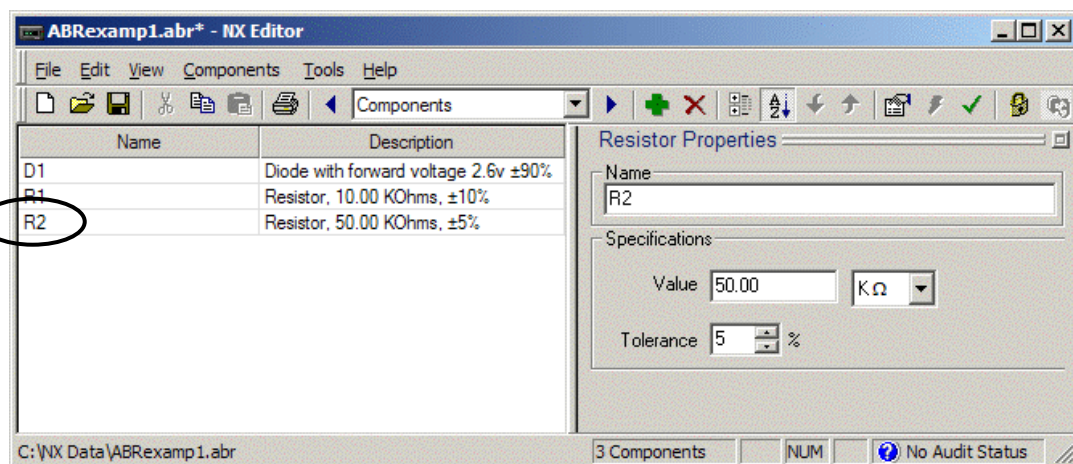
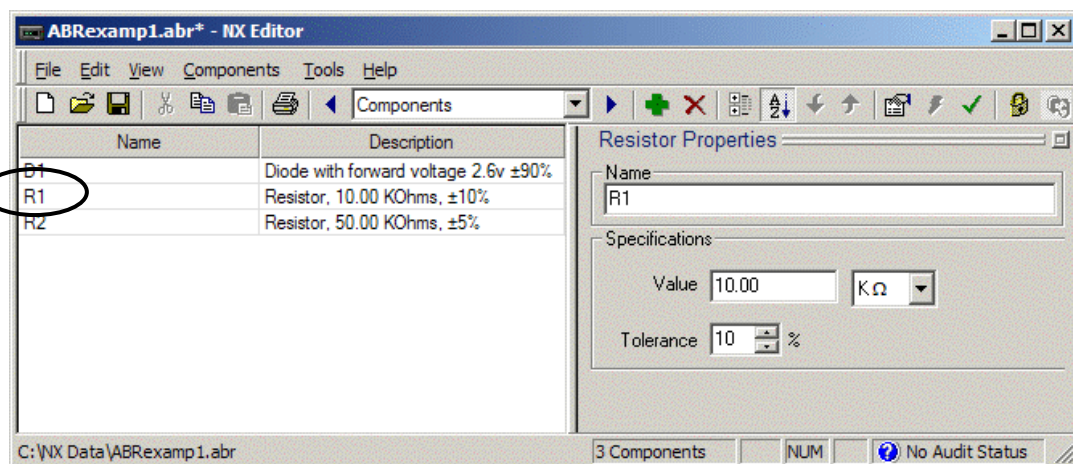
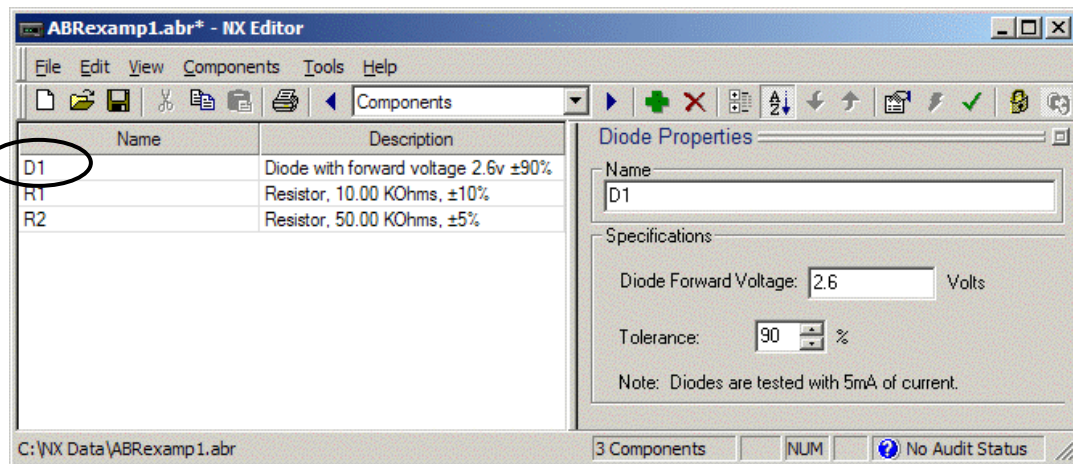
#### Step 2: Define the fixture blocks in the Fixture Blocks view

This task is identical to that which would be required for a normal NX Editor program. In this example, two fixture blocks (A and B) are defined with 4 pins each. Also, test points must be defined for each entry. For detailed information about defining Fixture Blocks and assigning test points, please refer to the *NX Editor User's Guide*. The result is shown below:



### Step 3: Define all components

This task is identical to that which would be required for a normal NX Editor program. In this example, two resistors and one diode are present in the overlays. For detailed information about defining components, please refer to the *NX Editor User's Guide*. The result is shown below:

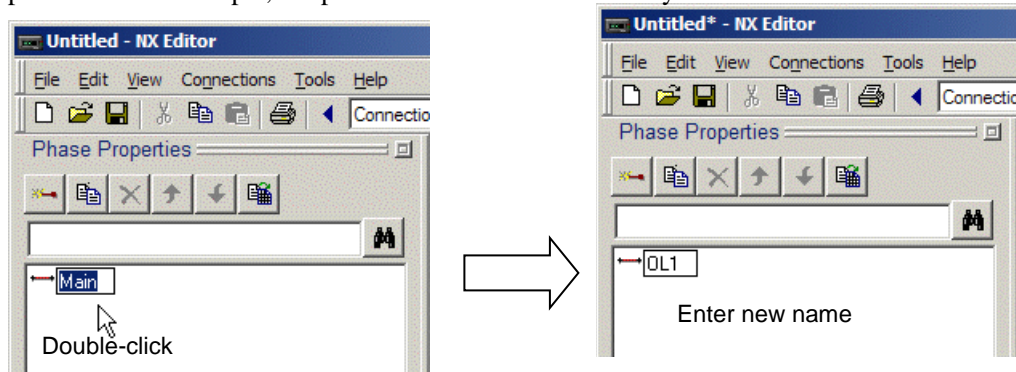


#### Step 4: Define a phase for each overlay

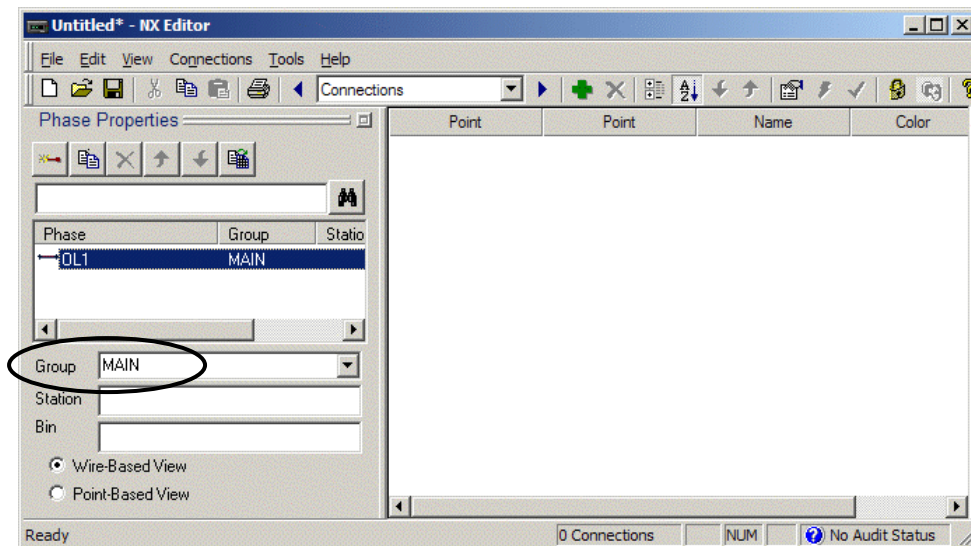
In the Connections View, a separate phase must be defined for each overlay, and the connections for each phase must be defined. Also, if the phase represents an overlay that is part of the base wire harness and is to be included as part of an overall shorts test of the wire harness, then the phase should be assigned to a common phase group. Phase Group names are arbitrary. In this example, 'MAIN' will be used to designate the Phase Group for overlays that comprise the base wire harness. (The "base wire harness" is the configuration of the wire harness before any post-assembly items are added. Post-assembly items are items such as shorting bars or relays that are added after the base harness is tested and that alter the electrical configuration of the wire harness. The "base wire harness" needs to be tested for shorts as well as for continuity – therefore, the overlays that make up this "base wire harness" must be grouped for shorts testing purposes).

For the first overlay: OL1, the following steps are required in the Connections View:

- Initially, the Connections View will have a pre-populated phase already defined named "Main". To change the name of this phase, double click on the name 'Main' and type in the desired name of the phase. In this example, the phase name for the first overlay is 'OL1'.

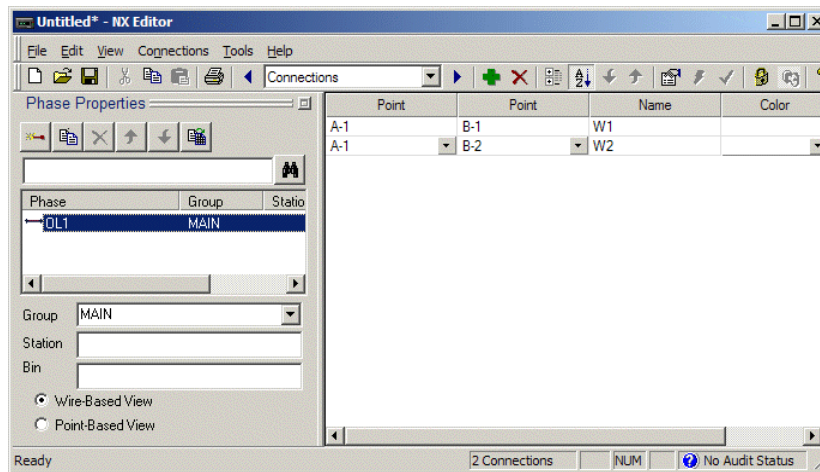


- Next, assign this Phase to a Phase Group. In this example, "MAIN" will be used as the Phase Group Name. To define Phase Group MAIN simply enter the name in the GROUP window as shown below:



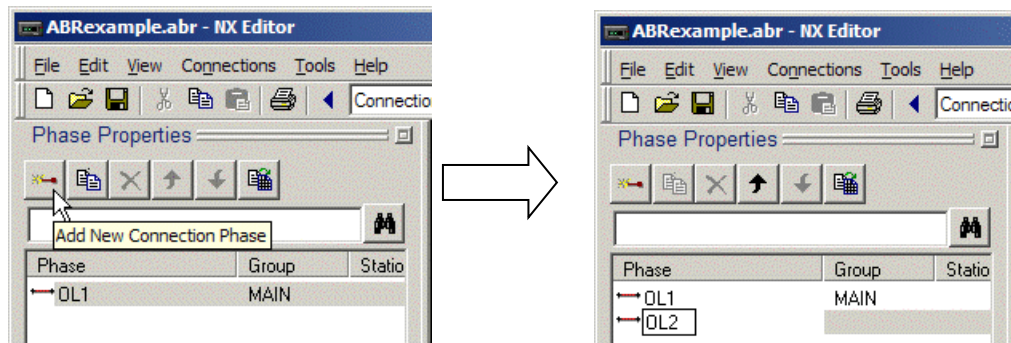


- Add all the connections for this overlay:

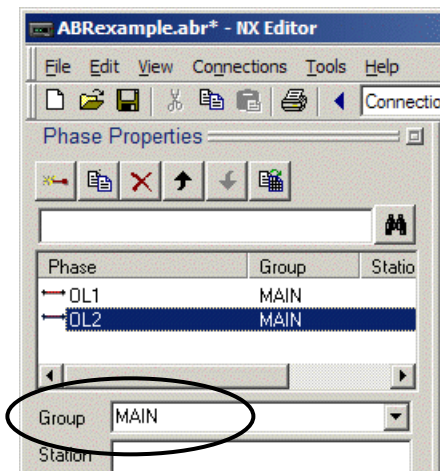


For the next overlay, the following steps are required in the Connections View:

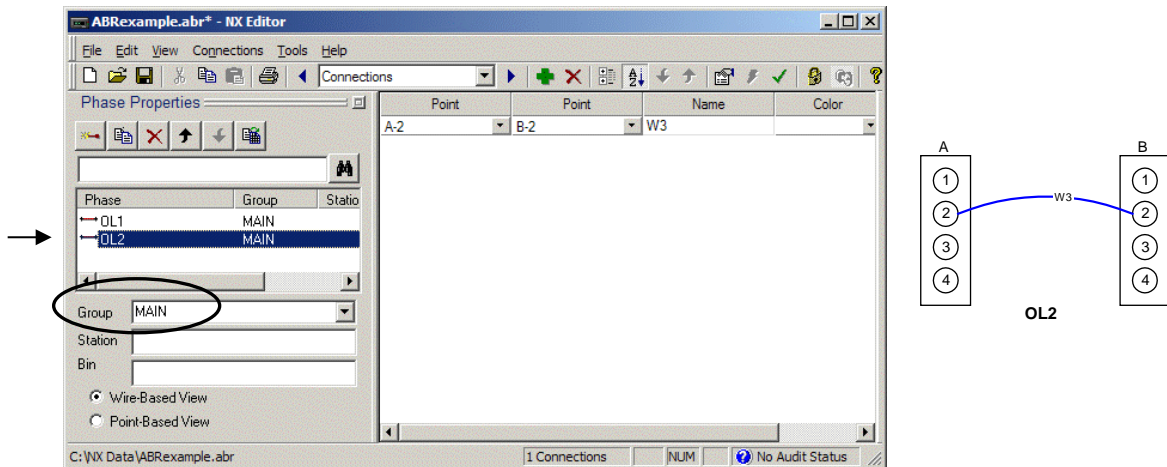
- In the Connections View, add a new phase and enter the phase name of the Overlay. The example below illustrates these steps for overlay OL2:



- Assign the newly defined phase to Phase Group MAIN:

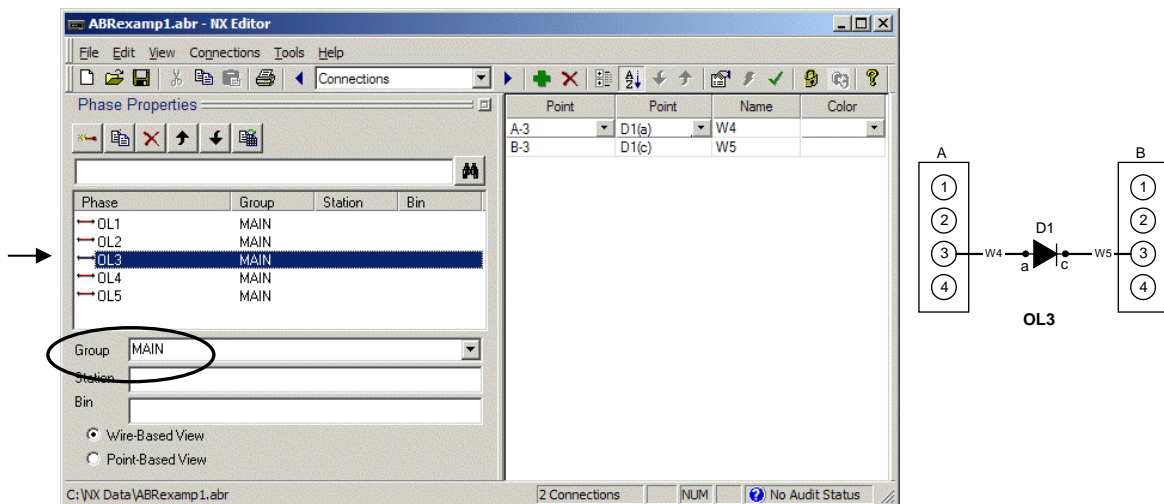


- Add all the connections for this overlay:

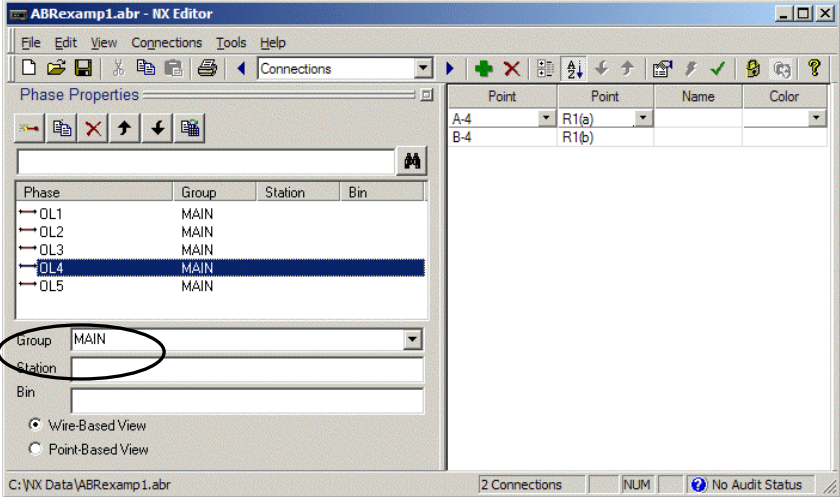


Repeat these steps for each remaining overlay. The following screen shots show the Connections View entries for each remaining overlay.

#### Overlay OL3:



## Overlay OL4:



Phase Properties

Phase	Group	Station	Bin
OL1	MAIN		
OL2	MAIN		
OL3	MAIN		
OL4	MAIN		
OL5	MAIN		

Group: MAIN

Station:

Bin:

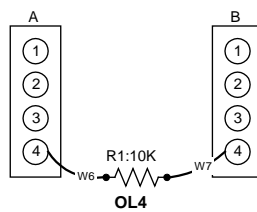
Wire-Based View

Point-Based View

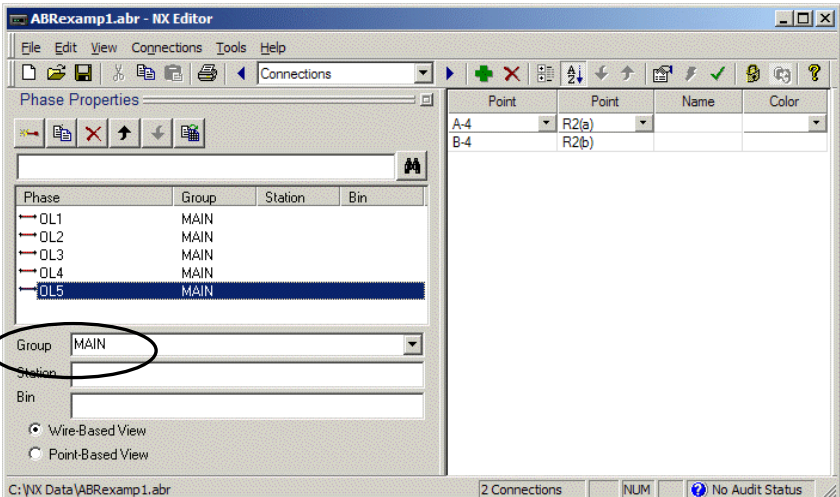
Connections

Point	Point	Name	Color
A-4	R1(a)		
B-4	R1(b)		

C:\NX Data\ABRexamp1.abr 2 Connections NUM No Audit Status



## Overlay OL5:



Phase Properties

Phase	Group	Station	Bin
OL1	MAIN		
OL2	MAIN		
OL3	MAIN		
OL4	MAIN		
OL5	MAIN		

Group: MAIN

Station:

Bin:

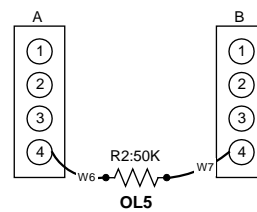
Wire-Based View

Point-Based View

Connections

Point	Point	Name	Color
A-4	R2(a)		
B-4	R2(b)		

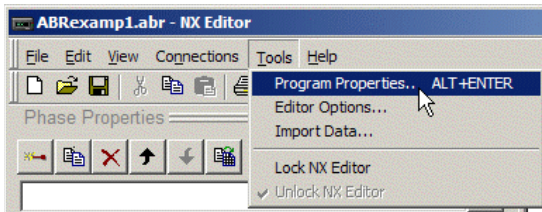
C:\NX Data\ABRexamp1.abr 2 Connections NUM No Audit Status



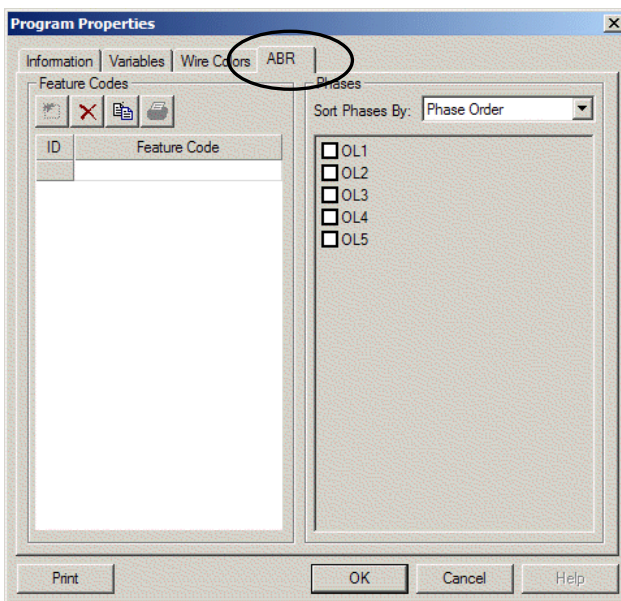
### Step 5: Define Feature Codes and associated Overlays

This step involves defining the feature codes and also defining the mapping of each feature code to its associated overlays.

To define Feature Codes and mappings to associated overlays, select Program Properties from the Tools menu:

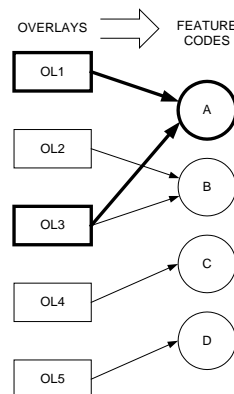
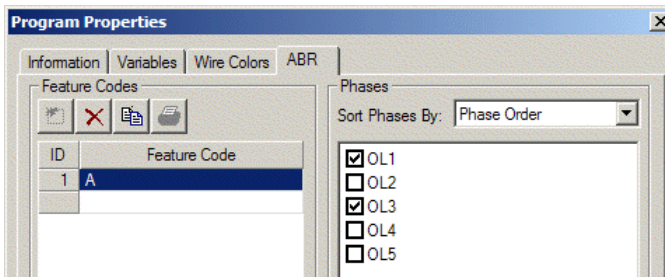


The Program Properties window will appear. Click on the ABR tab – the following screen will appear:



This window allows for the definition of Feature Codes and for associating one or more pre-defined overlays with each Feature Code. Simply enter the first Feature Code on the first line, press the Enter key, and then check the associated overlays as shown below:

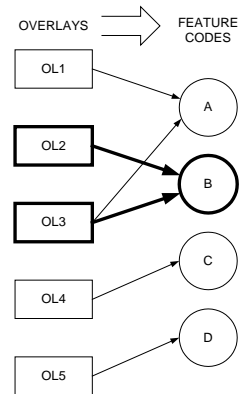
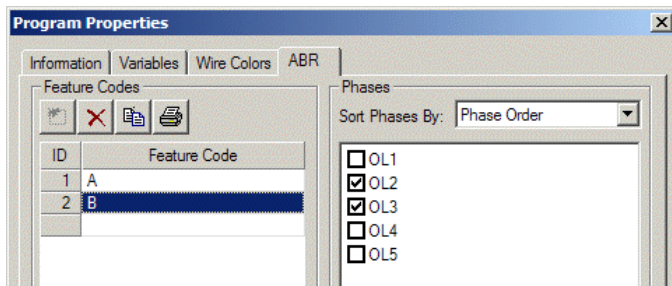
Feature Code A:



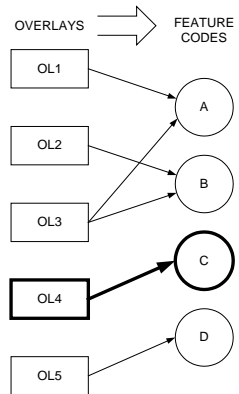
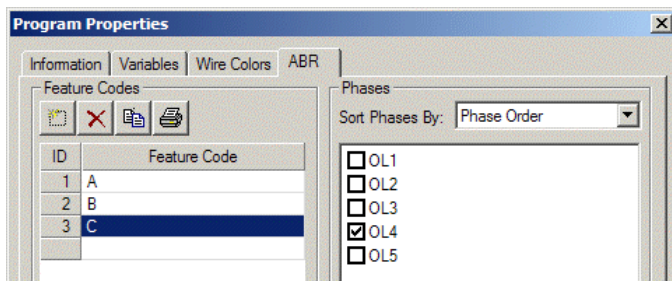


Repeat for each Feature Code as shown below:

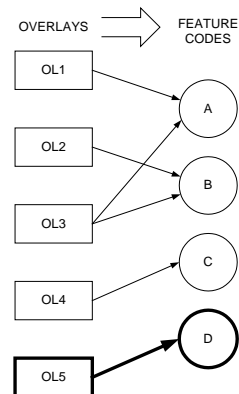
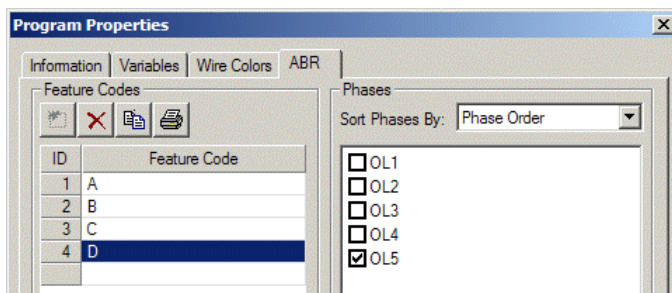
Feature Code B:



Feature Code C:



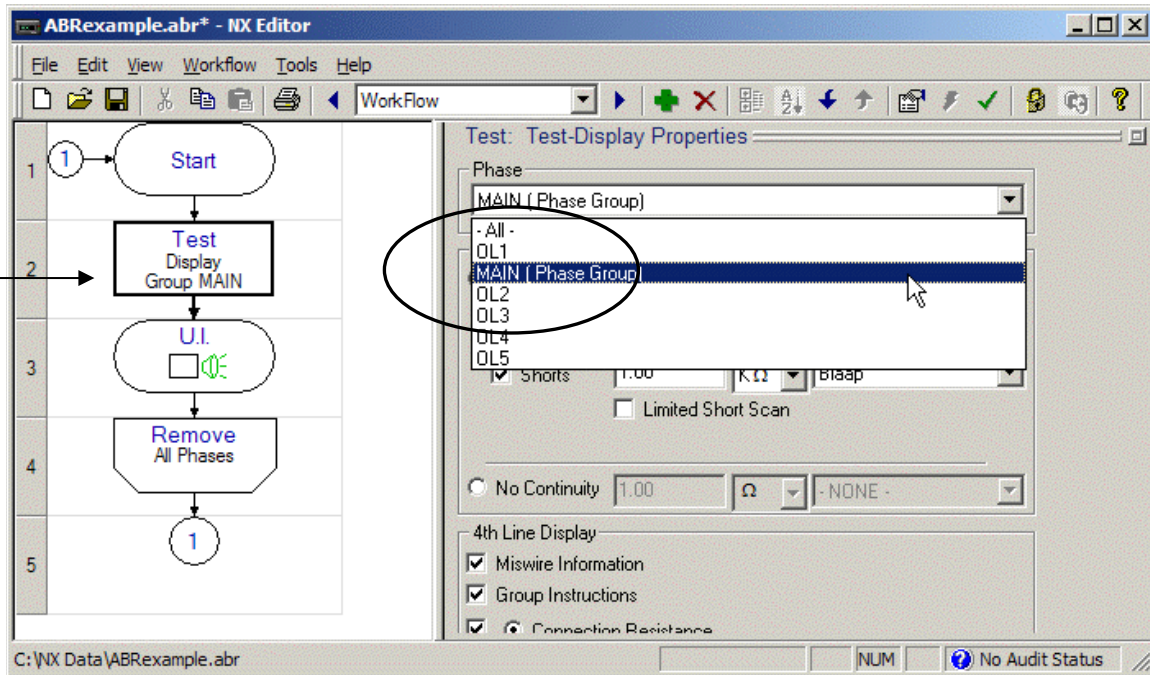
Feature Code D:



### Step 6: Modify the Workflow

In Step 4, each Overlay was assigned to a Phase Group. A Phase Group is a collection of Overlays that are to be combined for shorts testing. In this example, the Overlays were assigned to a Phase Group named “MAIN”.

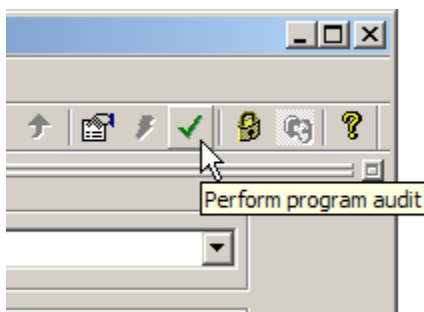
In the Workflow view, the Test – Display Workflow Item needs to be changed – select “MAIN (Phase Group)” as shown below:



By selecting Phase Group Main for the Test-Display Workflow item, all overlays that are assigned to this Phase Group will be combined for the purpose of shorts testing.

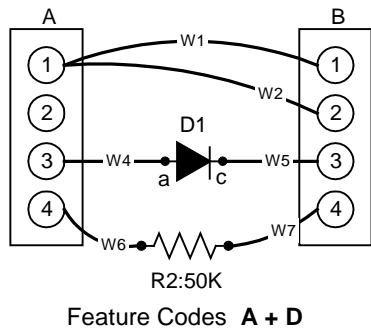
### Step 7: Audit the Program

Perform a program audit by pressing the audit button on the toolbar:

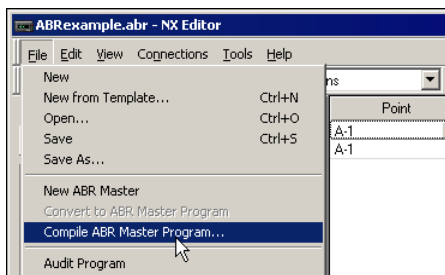


### Step 8: Compile a wire harness test program for a selected set of Feature Codes

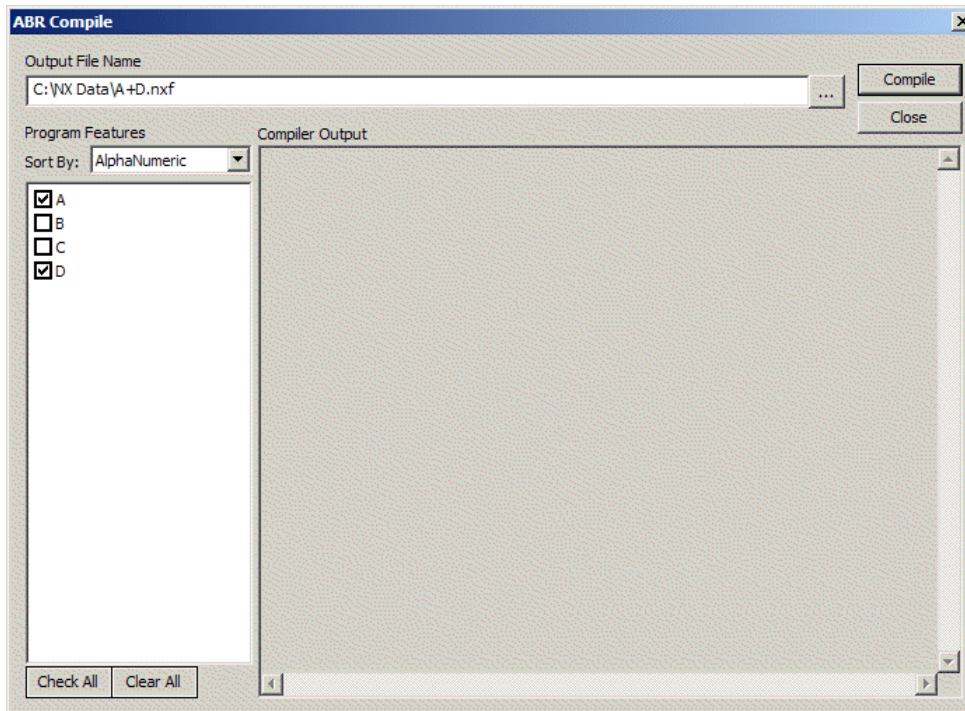
In this example, a test program will be compiled for the following wire harness, consisting of Feature Codes A and D:



From the File Menu, select “Compile ABR Master Program...”:

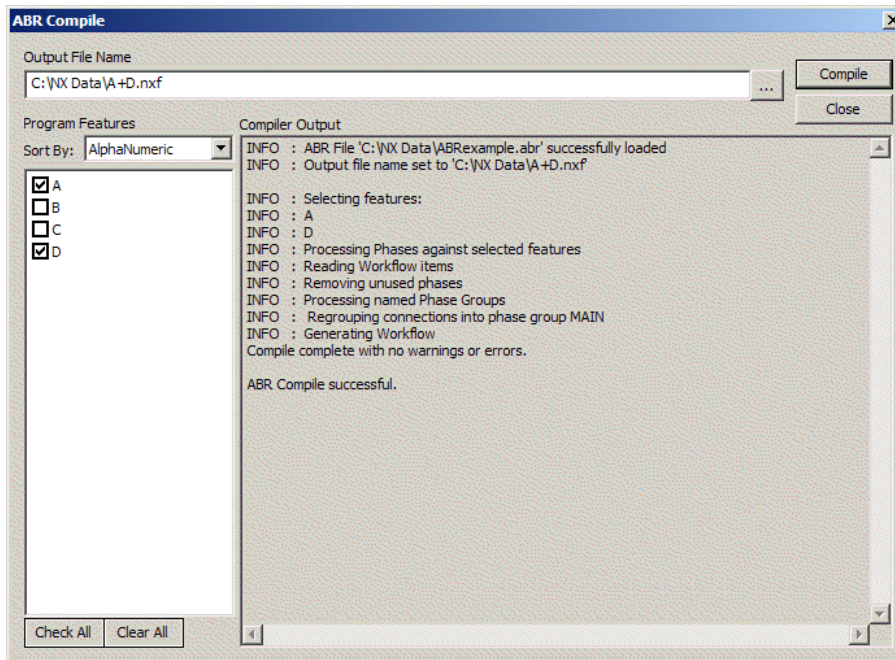


The ABR Compile window is displayed – it provides a checkbox for each defined Feature Code, and a input field for specifying the output file name – this is the compiled test program name. To compile, check the desired Feature Codes, enter the Output File Name, and press Compile:

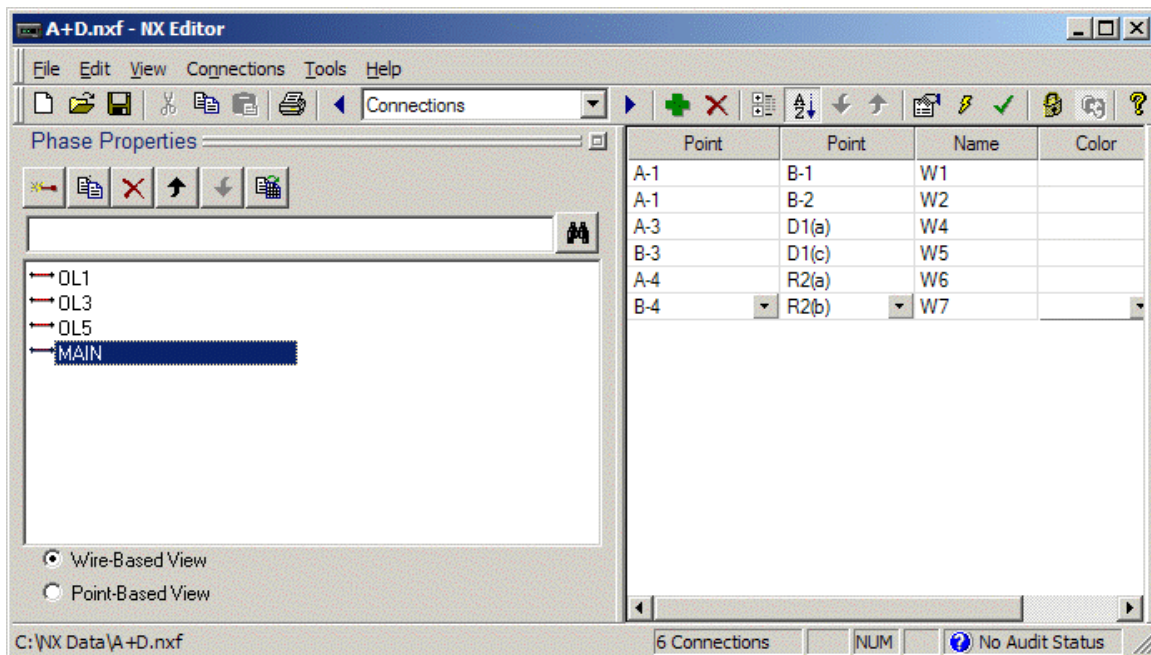




The NX Editor compiler will generate a test program for the selected Feature Codes – the results of this process are displayed in the “Compiler Output” window as shown below. The final line should be “ABR Compile successful”:

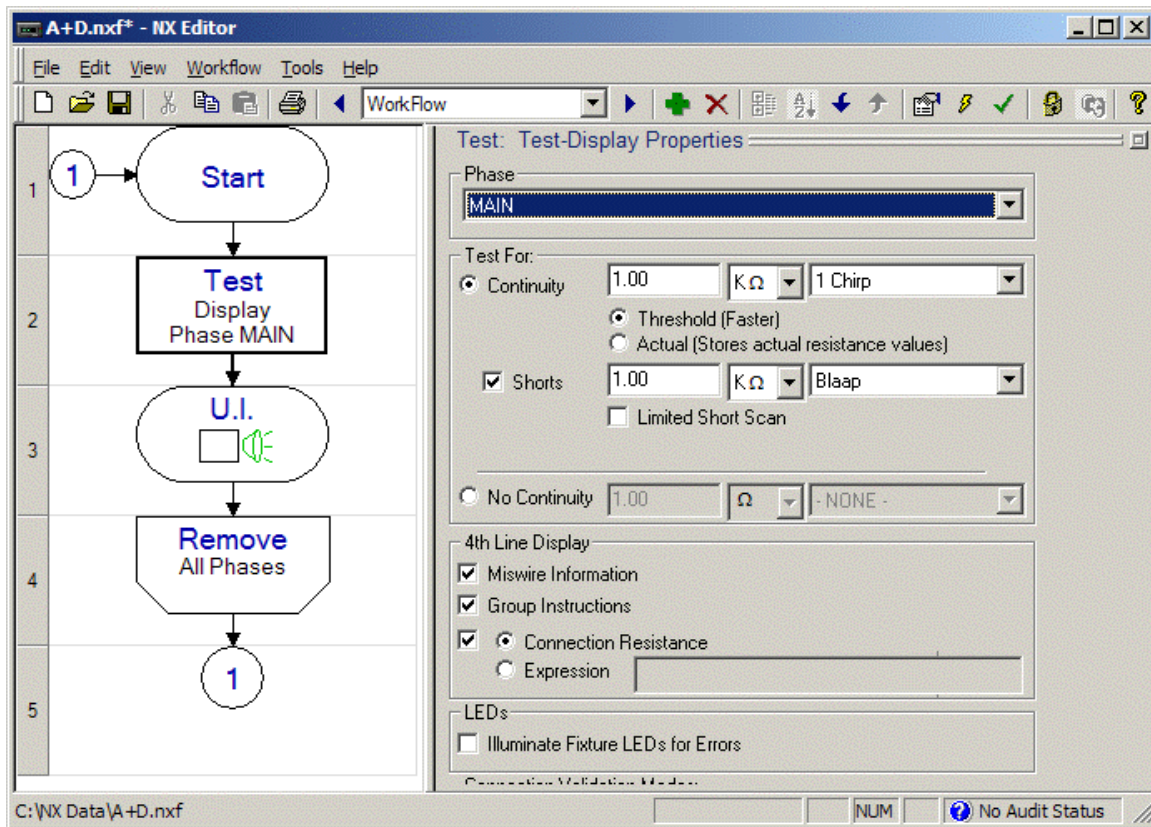


If the resulting compiled program, named “A+D.nxf” in this example, is opened with the NX Editor, the following is displayed in the Connections View:



Note that a Phase named “MAIN” has been created that contains all the connections defined in all the overlays associated with the selected Feature Codes. This is the result of having assigned these overlays to Phase Group “MAIN” in the master ABR file.

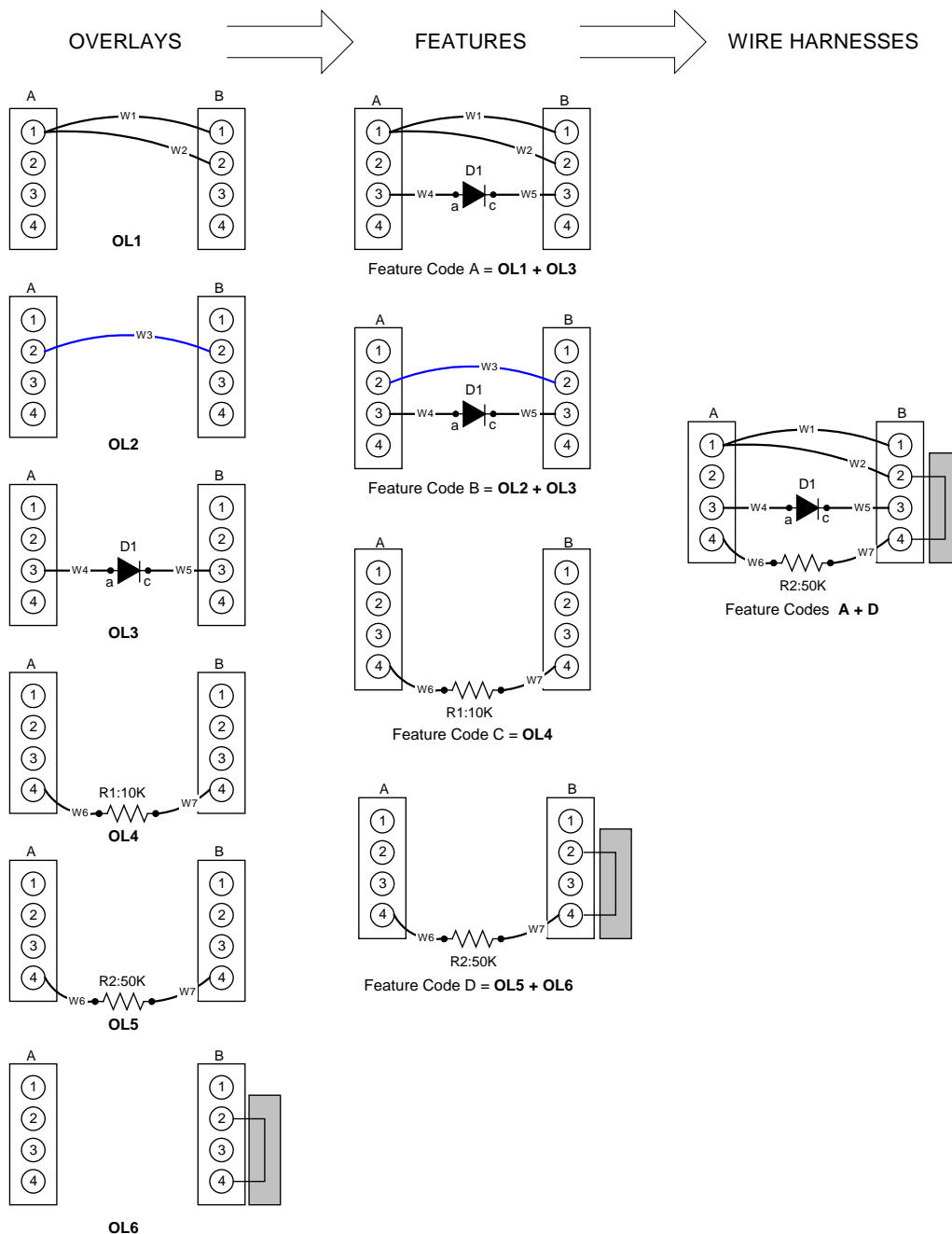
The following is displayed in the Workflow view of the compiled program:



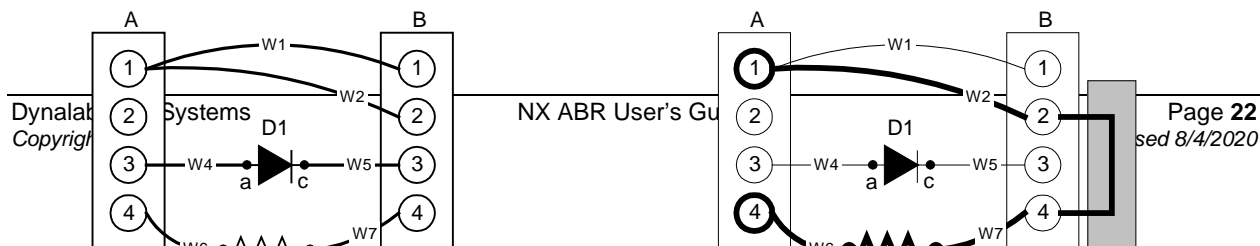
Note that Phase “MAIN” is being tested in the Test-Display Workflow item for Continuity and Shorts. This ensures that all the overlays associated with the selected Feature Codes are combined for the purpose of shorts testing.

## Testing a Wire Harness with Post-Assembled Parts

Post-assembled parts mate with a harness connector and alter the electrical configuration of the wire harness. Parts of this type include shorting caps, fuses, relay coils, and incandescent bulbs. The approach is to first test the wire harness without the part, then prompt the operator to add the part to the appropriate connector and test for electrical continuity through the pins on the harness that the part connects. Consider a modified version of the previous example:



Overlay 6 consists of a shorting cap that connects pins 2 and 4 of Connector B. It is defined to be part of Feature Code D.

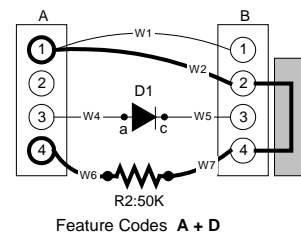
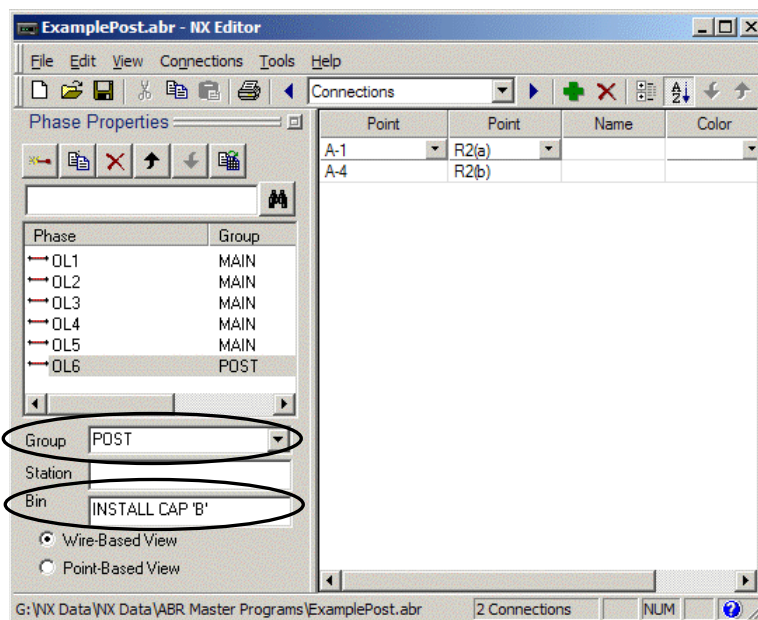


Step 2: Attach shorting cap and test  
for 50K resistance path between  
pins 1 and 4 of Connector A

The approach for testing a wire harness that has a post-assembled electrical part such as a shorting cap is a two step process as illustrated above. First, the harness is tested without the part for continuity and shorts. Then, the operator is prompted to attach the part to the appropriate connector, and continuity is tested through the path(s) created by the part. In the example above, adding the shorting cap creates a 50K ohm continuity path between pins 1 and 4 of Connector A.

The Master ABR file for this situation is similar to the previous example, with the following changes:

1 Add Overlay OL6 to the Connections View

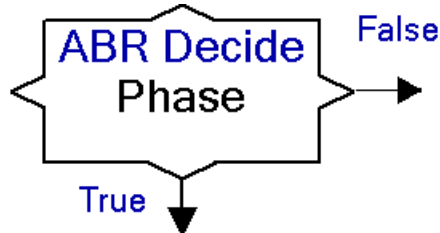


- Add a Phase named OL6 and enter the connections that are created by adding the post-assembled part.
- Assign all Phases for post assembled parts to a different Phase Group. In this example, Phase OL6 is assigned to Phase Group “POST”.
- Add an appropriate instruction in the field marked ‘Bin’. The contents of this field are displayed on the 4<sup>th</sup> line of the tester.

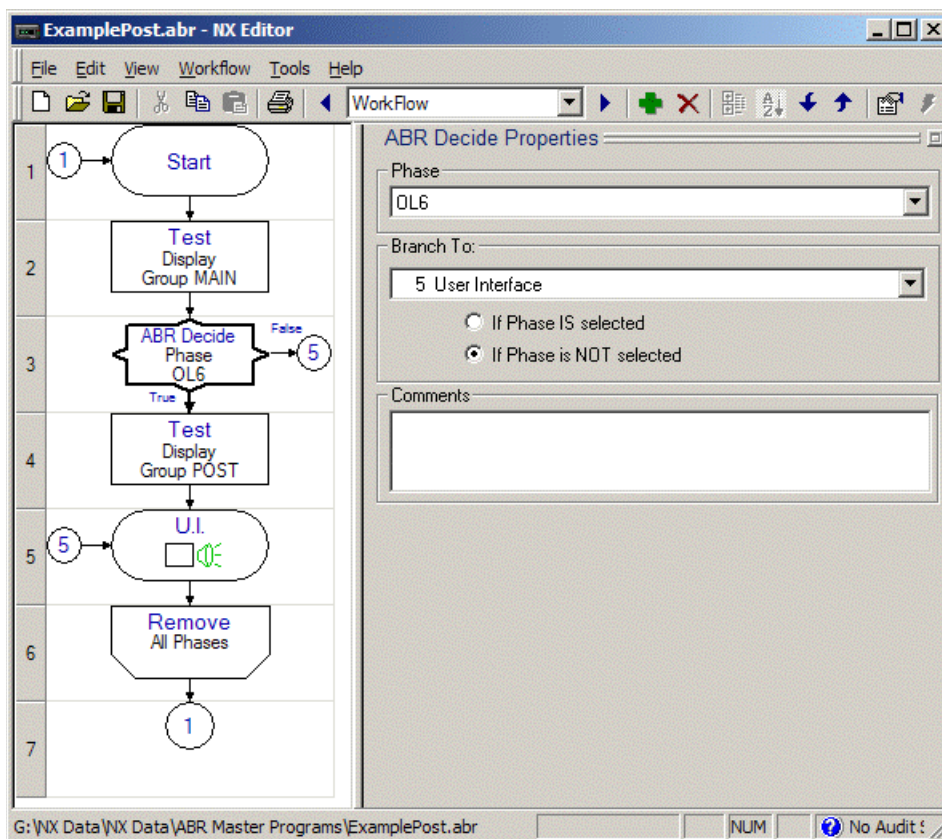


## 2 Modify the Workflow

Since not all wire harnesses will necessarily contain the Overlay that corresponds to a post-assembled item (in this example, OL6), it is necessary to have a branching capability in the workflow that is conditional upon the inclusion of a specific overlay in the compiled program. For this purpose, a workflow item named ABR Decide is available in the ABR Master.



The ABR Decide Workflow item enables branching based on the inclusion of a specific Phase. In this example, the ABR Decide Workflow item should appear as follows:



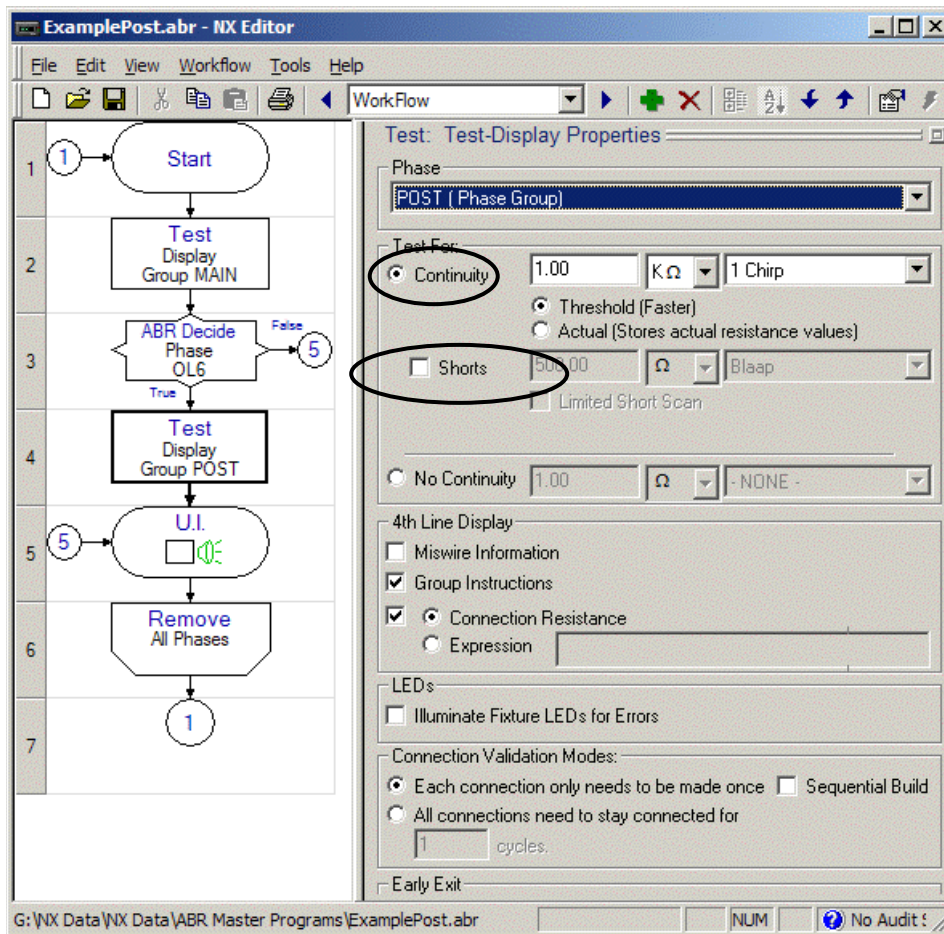
In this example, the ABR Decide appears after the initial Test of Group MAIN (this is the test of the wire harness before post assembly).

In Cell 3, the ABR Decide item checks to see if the Phase named "OL6" is included in set of Feature Codes selected for compilation. If it is included, the workflow proceeds to cell 4 – this is a Test of Group POST – in this example, POST is the PhaseGroup to which phase OL6 was assigned. Cell 4 tests for the presence of the shorting cap.

If Phase OL6 is not included, then the workflow proceeds to cell 5, skipping over the Test for the shorting cap.



The Test workflow item in cell 4 is shown below:



Note that in the Properties pane, the checkbox next to Shorts is empty and the checkbox next to Miswire Information is empty. It is important that both of these boxes are un-checked.

Cell 5 is a User Interface cell indicating that the harness is good - Cell 6 tests for removal of the wire harness.

To illustrate this example further, after Cell 2 is complete (main wire harness has been tested), if overlay OL6 is included, the tester will display the following:

R2	1
A-1	
A-4	
INSTALL CAP 'B'	

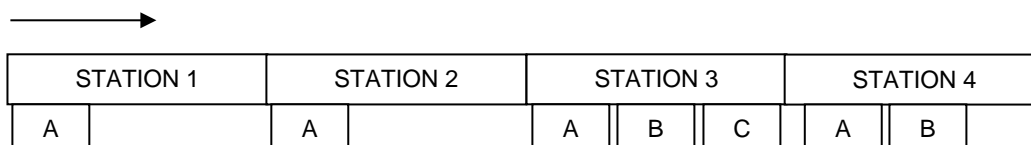
# Progressive Assembly Example

The examples shown in Chapter 2 are based on testing a completed assembly. However, the NX ABR feature also supports the concept of progressively assembling the wire harness on an assembly line. There are two major differences in this approach as compared to the previous examples:

- 1 While the wire harness is being assembled, no shorts testing is performed – only continuity testing is done.
- 2 After the base harness has been assembled, complete continuity and shorts testing is done.
- 3 Parts such as shorting caps are assembled and a continuity test is done to check for the presence of the part.
- 4 Each overlay phase may be associated with a Station ID and a Bin ID. This supports the concept of parts bins at each station. The NX ABR Connections View allows for the identification of the Station and Bin for each overlay.

Again, the most effective way of explaining the method of building an ABR master program for a wire harness that will be assembled progressively is through the use of an example:

Consider a progressive assembly line consisting of four stations. Stations 1 and 2 each has one parts bin labeled A. Station 3 has three parts bins: A, B, and C. Station 4 has two parts bins: A and B.



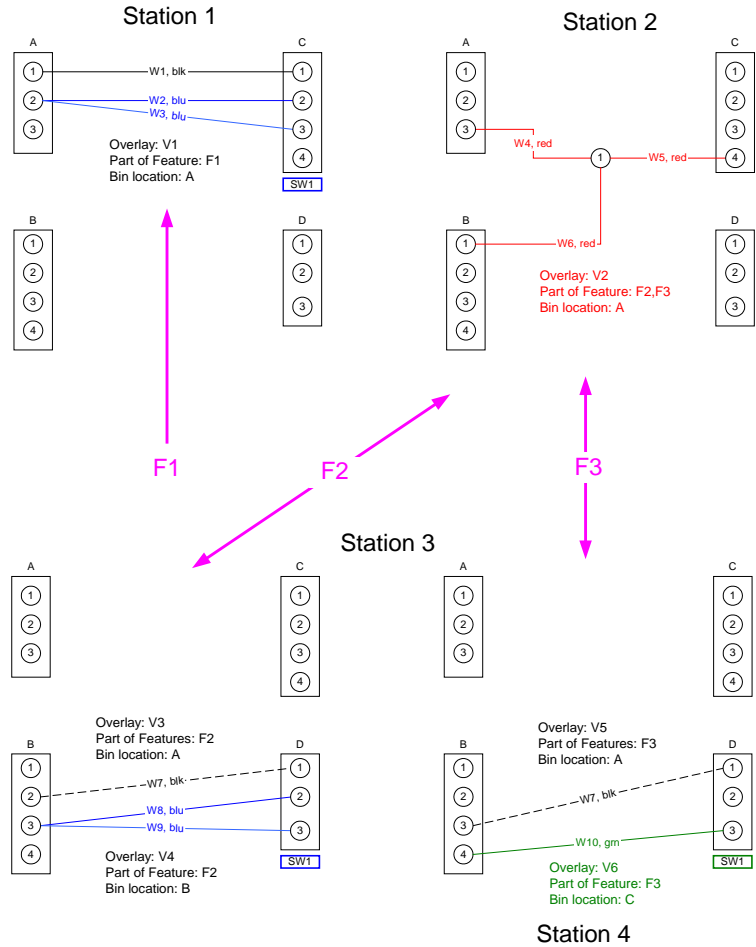
The following example illustrates a 4-connector wire harness with three Feature Codes (F1, F2, and F3), consisting of eight overlays (V1 – V8) to be progressively assembled on an assembly line configured as shown above.

Overlay V1, consisting of wires W1, W2, and W3 is assembled at Station 1. It is part of Feature Code F1. Parts are in Bin A.

Overlay V2, consisting of wires W4, W5, and W6 is assembled at Station 2. It is part of Feature Codes F2 and F3. Parts are in Bin A.

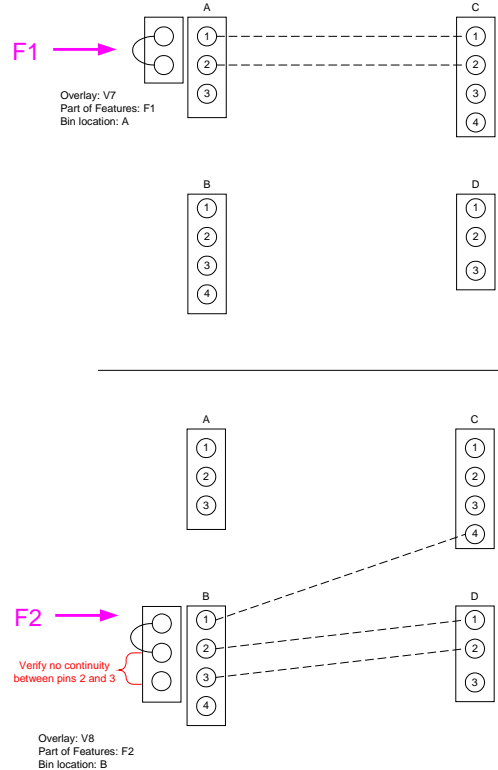
Overlay V3, consisting of wire W7, and Overlay V4, consisting of wires W8 and W9 are assembled at Station 3. They are part of Feature Code F2. Parts for Overlay V3 are in Bin A. Parts for Overlay V4 are in Bin B.

Overlay V5, consisting of wire W7, and Overlay V6, consisting of wire W10 are also assembled at Station 3. They are part of Feature Code F3. Parts for Overlay V5 are in Bin A. Parts



Overlay V7, consisting of a shorting cap is assembled after the base wire harness has been tested (post-assembly part). It is assembled at Station 4 and is part of Feature Code F1. Parts for Overlay V7 are located in Bin A.

Overlay V8, consisting of a shorting cap is assembled after the base wire harness has been tested (post-assembly part). It is assembled at Station 4 and is part of Feature Code F2. Parts for Overlay V8 are located in Bin B.

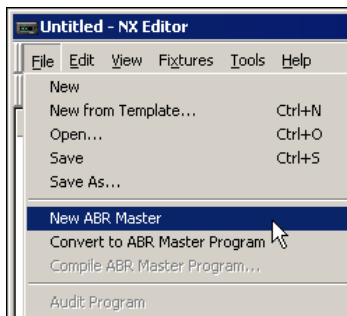


The following is a list of the tasks required to implement this example, using the NX Editor with the ABR feature enabled:

- 1 Create a new ABR master file
- 2 Define the fixture blocks in the Fixture Blocks view.
- 3 Define all components (one splice) in the Components view.
- 4 In the Connections View, define a phase for each overlay, associate each phase with an appropriate Phase Group, and define all connections for each phase.
- 5 Define each Feature Code and define the associated overlays for each Feature Code – this is done in the Program Properties -> ABR window
- 6 Modify the Workflow
- 7 Audit the program
- 8 Compile a wire harness test program for a selected set of Feature Codes

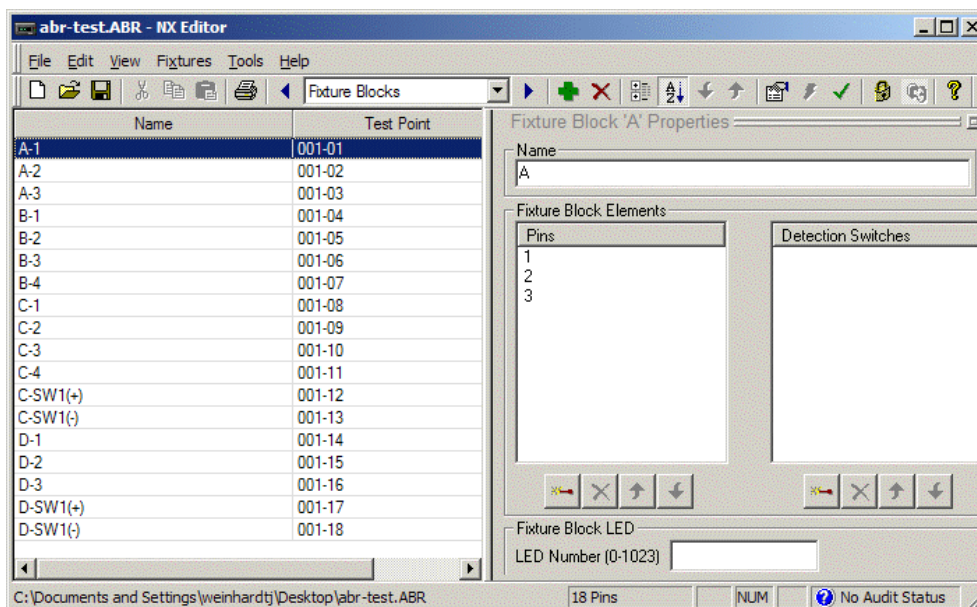
### Step 1: Create new ABR master file

From the File menu, select “New ABR Master”. This enables the ABR features.



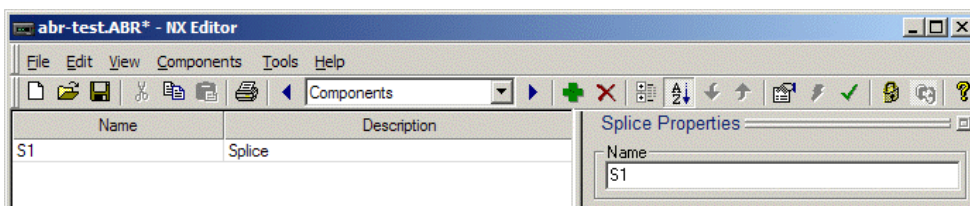
### Step 2: Define the fixture blocks in the Fixture Blocks view

This task is identical to that which would be required for a normal NX Editor program. In this example, four fixture blocks (A, B, C, & D) are defined. A detection switch is defined for Connectors C and D. Also, test points must be defined for each entry. The result is shown below:



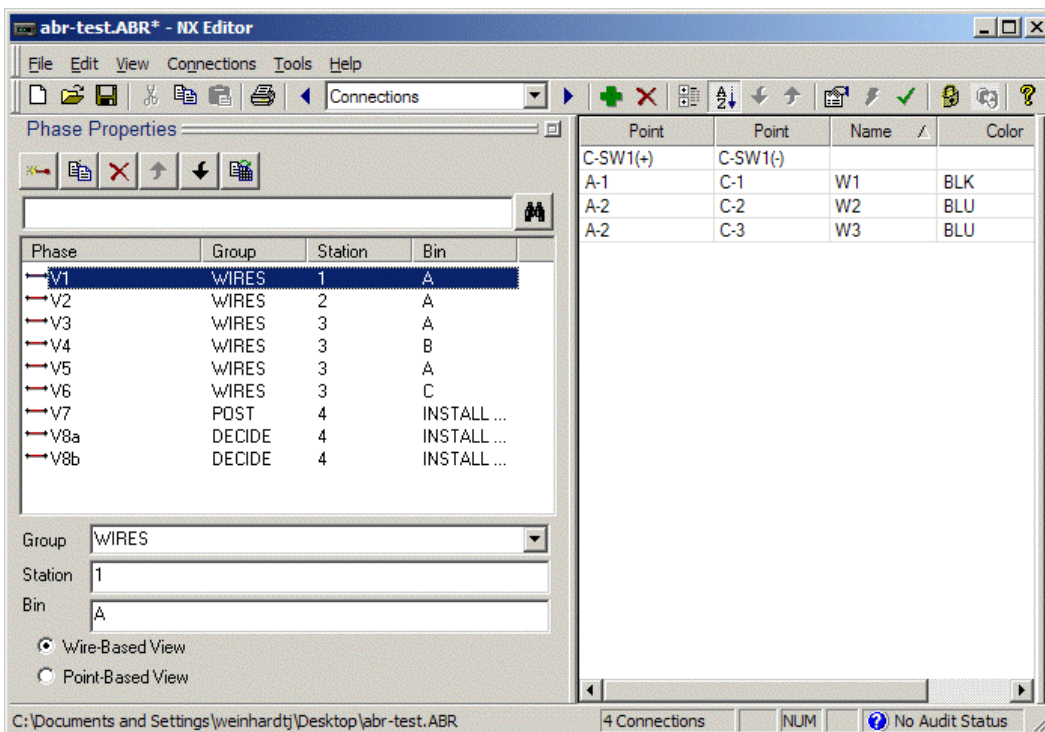
### Step 3: Define all components

This task is identical to that which would be required for a normal NX Editor program. In this example, a splice is present in the overlays. For detailed information about defining components, please refer to the *NX Editor User's Guide*. The result is shown below:



### Step 4: Define a phase for each overlay

In the Connections View, a separate phase must be defined for each overlay, and the connections for each phase must be defined. Also, if the phase represents an overlay that is part of the base wire harness and is to be included as part of an overall shorts test of the wire harness, then the phase should be assigned to a common phase group. Phase Group names are arbitrary. In this example, 'WIRES' will be used to designate the Phase Group for overlays that comprise the base wire harness. (The "base wire harness" is the configuration of the wire harness before any post-assembly items are added. Post-assembly items are items such as shorting bars or relays that are added after the base harness is tested and that alter the electrical configuration of the wire harness. The "base wire harness" needs to be tested for shorts as well as for continuity – therefore, the overlays that make up this "base wire harness" must be grouped for shorts testing purposes). Phases associated with post assembled parts should be assigned to a unique Phase Group. In this example, "POST" will be used for this group name. Finally, any overlays for post-assembled parts that require additional workflow control should be assigned to the same Phase Group name. In this example, "DECIDE" will be used for this group name.



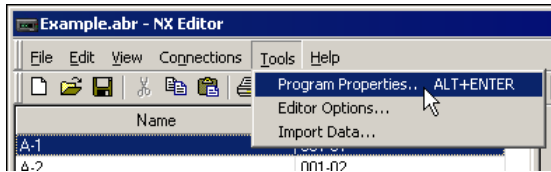
Note: For post-assembled items, the Bin field contains the instruction for the operator that is to appear on the 4<sup>th</sup> line of the display.



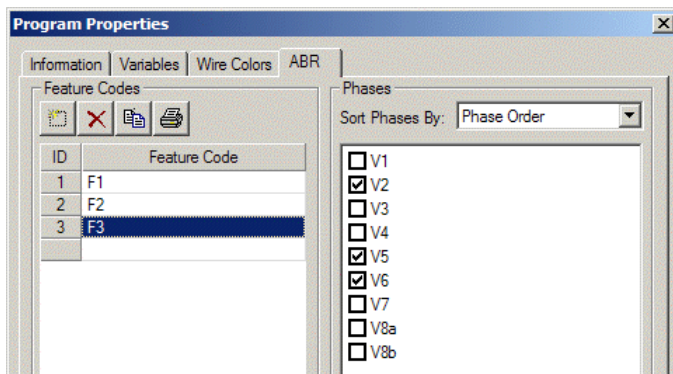
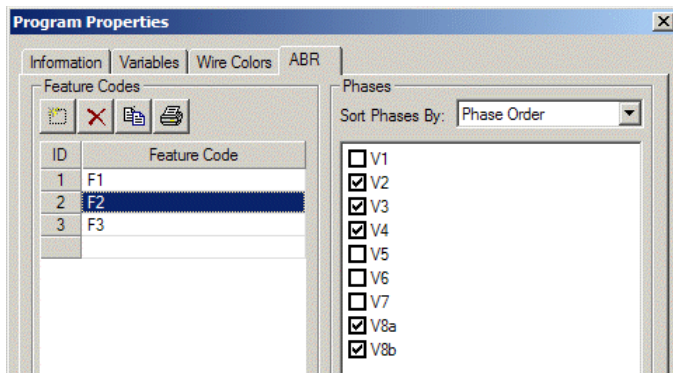
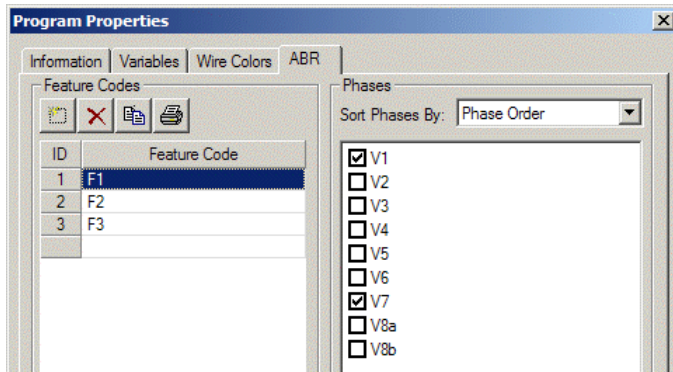
### Step 5: Define Feature Codes and associated Overlays

This step involves defining the feature codes and also defining the mapping of each feature code to its associated overlays.

To define Feature Codes and mappings to associated overlays, select Program Properties from the Tools menu:

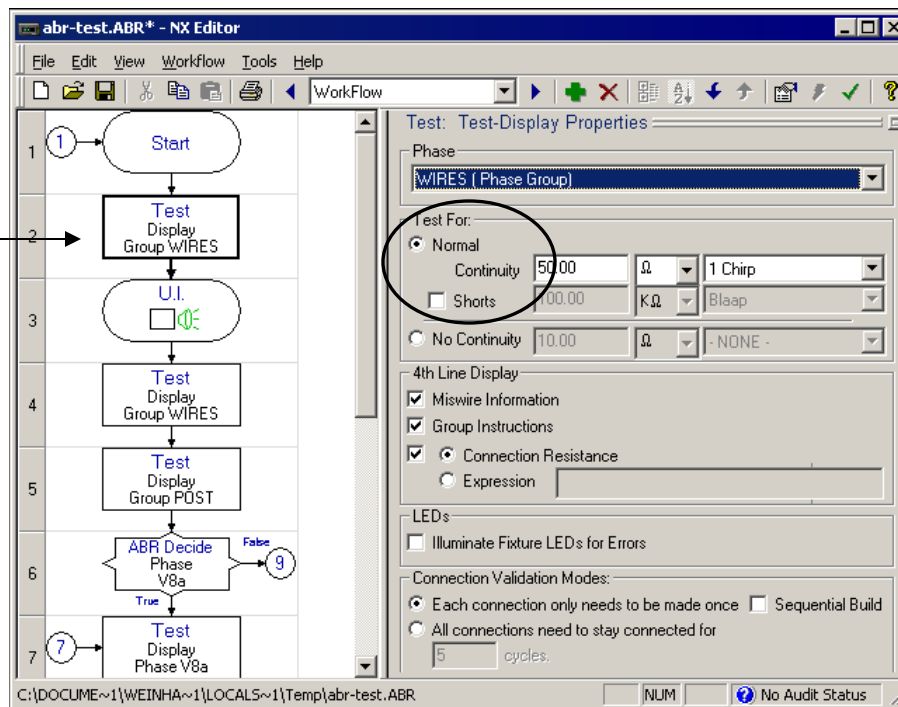


The Program Properties window will appear. Click on the ABR tab and complete the mapping for each Feature Code as illustrated below:

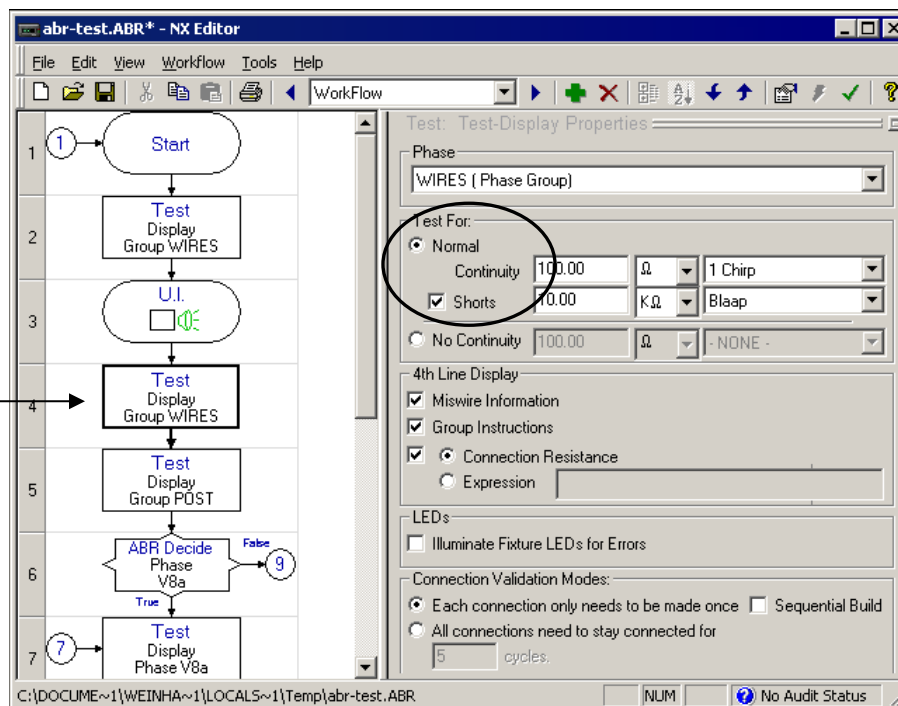


## Step 6: Modify the Workflow

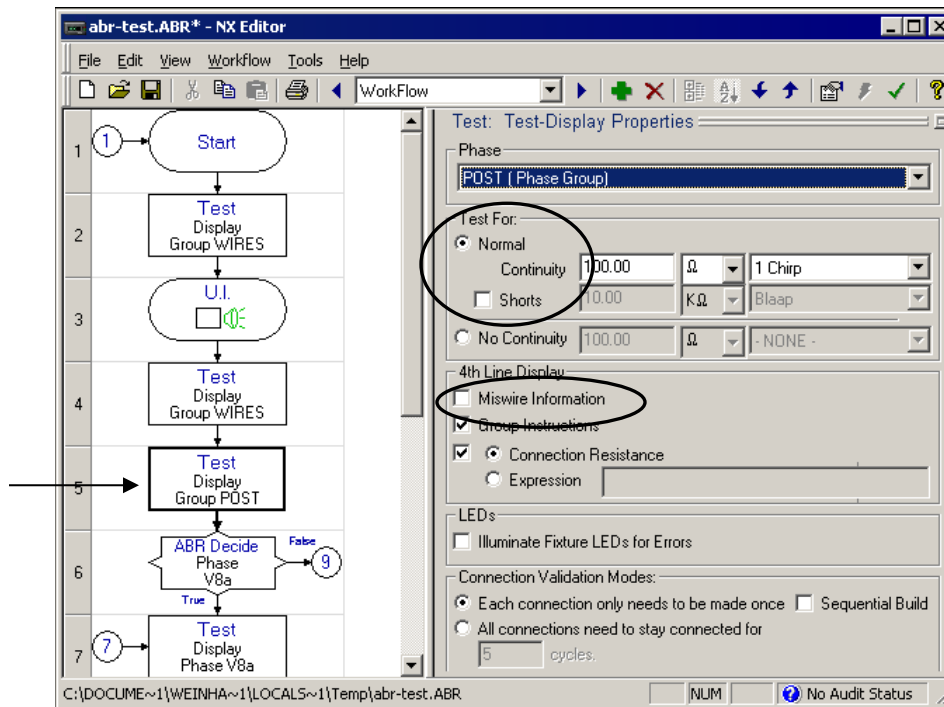
Cell 2 tests for the Overlays as they are being progressively assembled – continuity only testing is done (Shorts box is not checked).



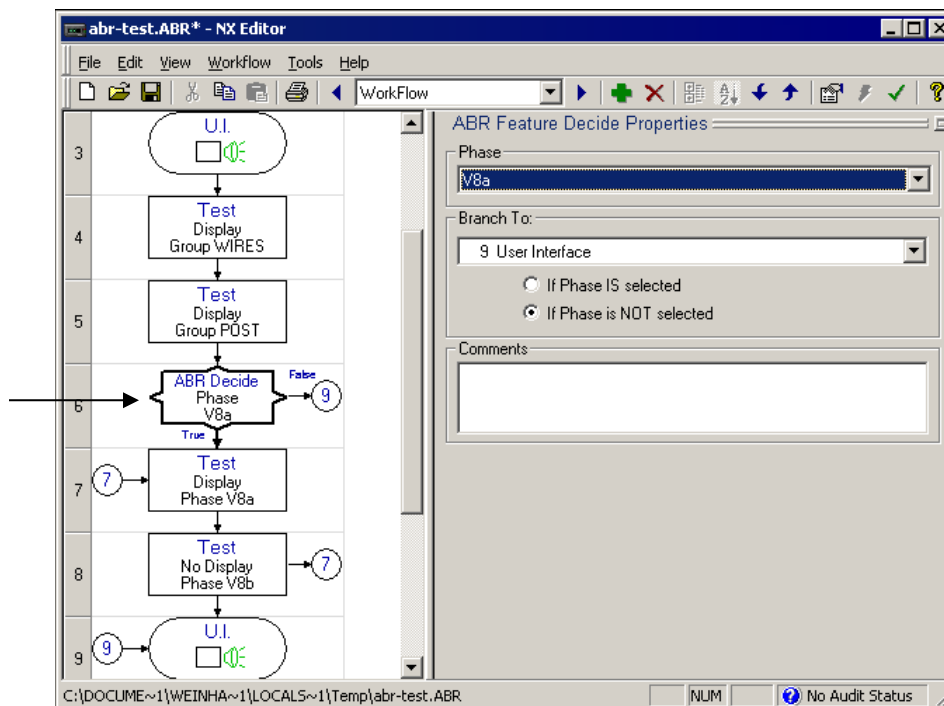
Cell 4 does a complete continuity & shorts test for the completely assembled base harness.



Cell 5 does a continuity test only for the presence of all unconditional post-assembled items. The Shorts box should not be checked. The Miswire Information box should not be checked.

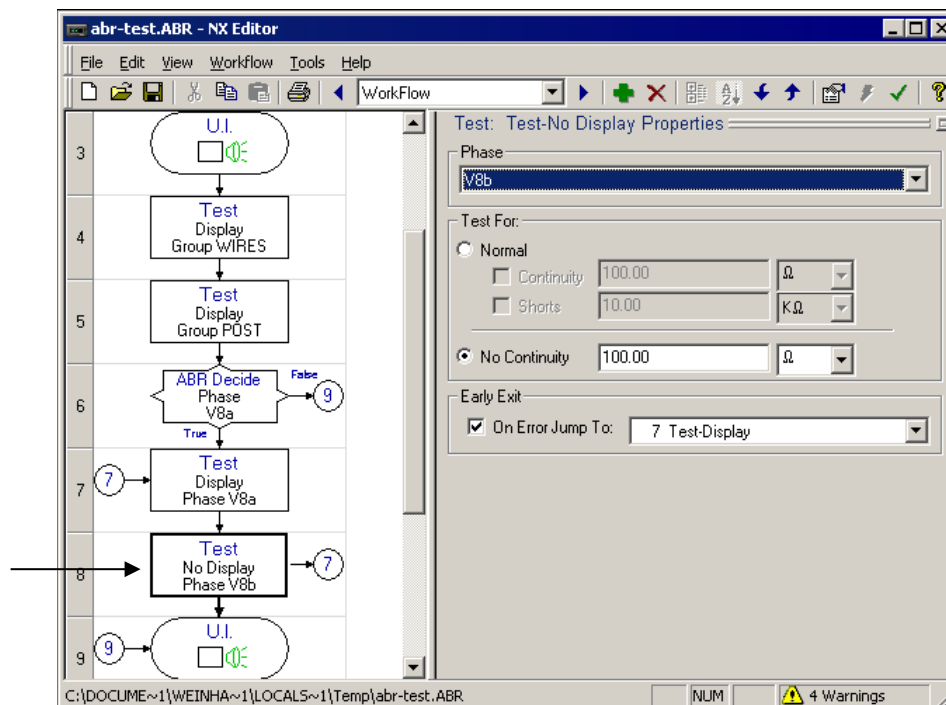
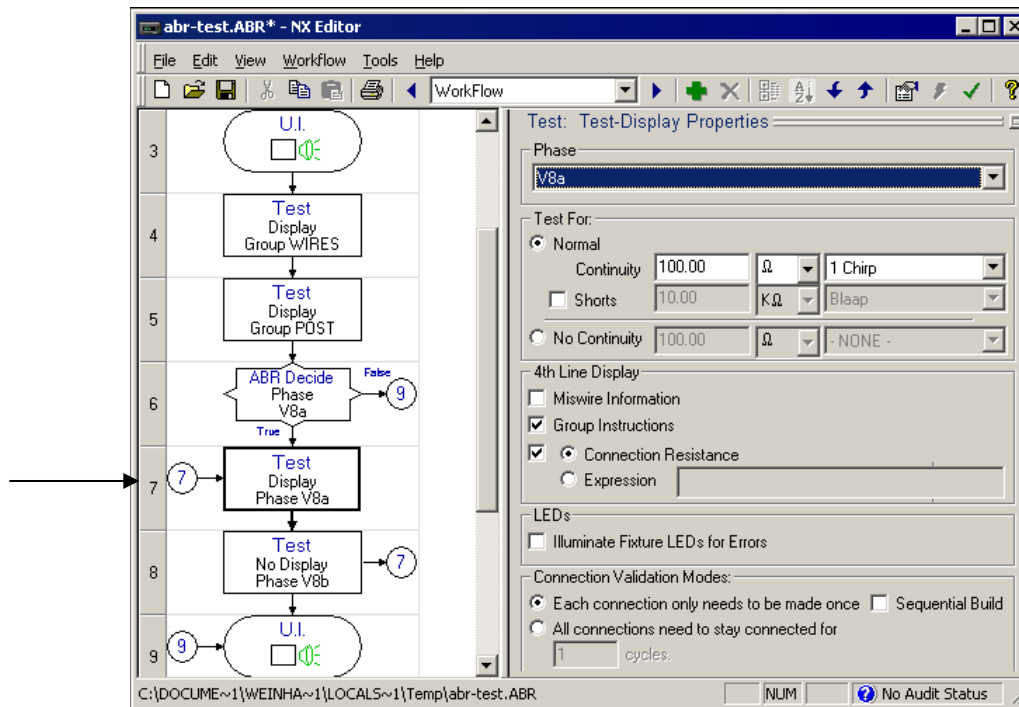


Cell 6 is the ABR Decide Workflow item. This is used for conditional post-assembled parts. The ABR Decide item branches based on the inclusion of a phase (or overlay). In this case, the decision is based upon the inclusion of overlay V8a.



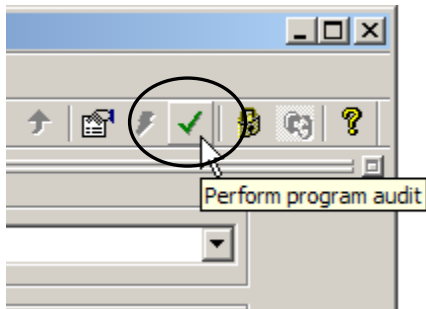


Cells 7 and 8 combine to simultaneously check for the presence of the shorting cap that connects pins 1 and 2 of Connector B, and for no continuity across pins 2 and 3 of Connector B.

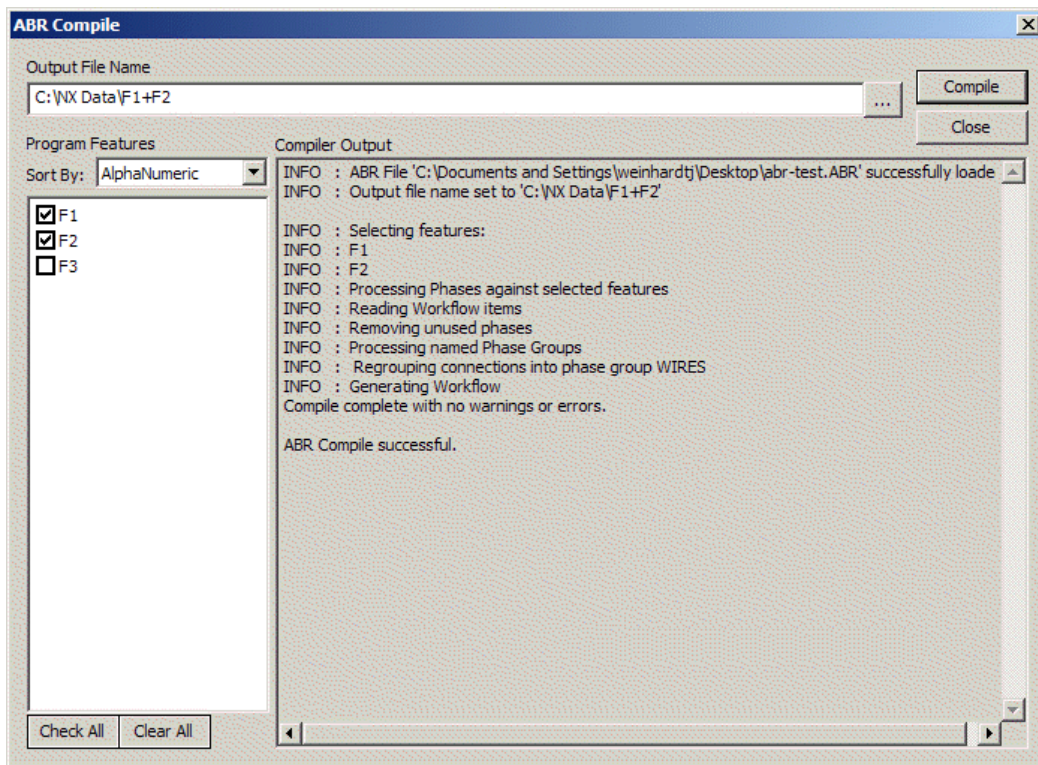
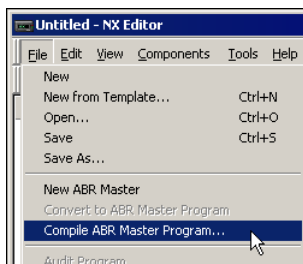


### Step 7: Audit the Program

Perform a program audit by pressing the audit button on the toolbar:



### Step 8: Compile a wire harness test program for a selected set of Feature Codes



# **NX Editor ABR Enabled Software License Agreement**

## **DEFINITIONS:**

The term "Software" as used in this AGREEMENT means the computer programs, together with any updates or conversions subsequently supplied by Dynalab Test Systems, Inc.

The term "Eligible Analyzer Products" means any analyzer sold by Dynalab Test Systems, Inc.

## **TITLE AND OWNERSHIP**

The Software, including the original and any copies, and any copyright, and trade secrets remain the property of Dynalab Test Systems, Inc. You agree to protect the Software from unauthorized use, reproduction, distribution, or publication. All rights not specifically granted in this license are reserved by Dynalab Test Systems, Inc.

## **PERMITTED USES**

You may:

- \* Install and use multiple copies for the purpose to create ABR master files
- \* Make copies of the Software for back-up or archival purposes.

## **USES NOT PERMITTED**

You may not:

- \* Use the Software for any purposes unless this AGREEMENT is entered into.
- \* Make copies of the Software, except as stated in the "Permitted Uses" section of this AGREEMENT.
- \* Rent, lease, sub-license, time-share, lend, or transfer the Software or your rights under this License without the prior written authorization of Dynalab Test Systems, Inc.
- \* Alter, decompile, disassemble, or reverse engineer the Software.
- \* Remove, alter, or obscure the Dynalab Test Systems, Inc., copyright and/or trademark notices.
- \* Use the Software to generate programs for products that are not Eligible Analyzer Products.

## **WARRANTY AND LIMITATION OF LIABILITY**

Dynalab Test Systems, Inc., warrants that the Software will perform substantially for a period of ninety (90) days from the date of receipt.

**CUSTOMER REMEDIES:** Dynalab Test Systems, Inc.'s liability and your exclusive remedy shall be, at Dynalab's option, either: return of the purchase price, or repair or replacement of the Software. This limited warranty is void if failure of the Software has resulted from accident, abuse, or misapplication. Any replacement Software will be warranted for the remainder of the original warranty period.

NO OTHER WARRANTIES: DYNALAB DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE, OR ANY ACCOMPANYING WRITTEN MATERIALS. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE TO STATE.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES: IN NO EVENT SHALL DYNALAB BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS BUSINESS, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF DYNALAB HAS BEEN MADE AWARE OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT RECOGNIZE THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

#### TERM

Without prejudice to any other rights, Dynalab Test Systems, Inc. may terminate this AGREEMENT if you fail to comply with the terms and conditions of this AGREEMENT. In such event, you must return the Software and all of its component parts with all copies, modifications and merged portions in any form to Dynalab Test Systems, Inc.

#### GENERAL

This AGREEMENT represents our entire understanding and agreement regarding the Software and supersedes any prior written or oral agreement, purchase order, license, communications, advertising, or representations.

If any provisions of this AGREEMENT shall be unlawful, void, or for any reasons unenforceable, it shall be deemed severable from, and shall in no way affect the validity or enforceability of the remaining portions of this AGREEMENT. This AGREEMENT will be governed by the laws of the State of Ohio.