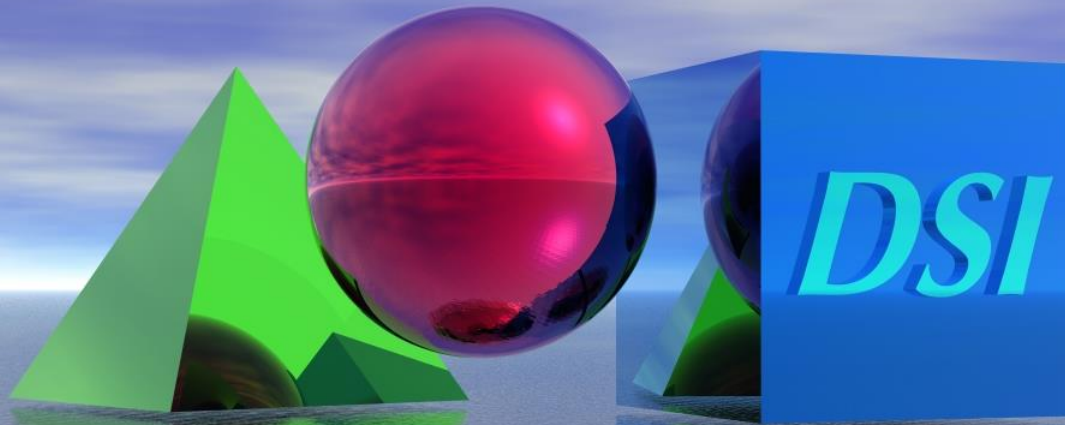


# The *eXpress* Maintenance Module



DSI International  
November, 2015

# The *eXpress* Maintenance Module

The *eXpress* Maintenance Module provides a set of features that allow you to extend diagnostics created within *eXpress* to address a variety of practical situations:

- Transferal of Knowledge between Different Levels of Diagnostics
- Coordinated Development of Diagnostic & Maintenance Procedures
- Concurrent Engineering of Multiple-Level Diagnostics
- Implementation of Segmented Diagnostics
- Customization / Optimization of Repair Procedures (**Spring 2016**)

# Transferal of Diagnostic Knowledge

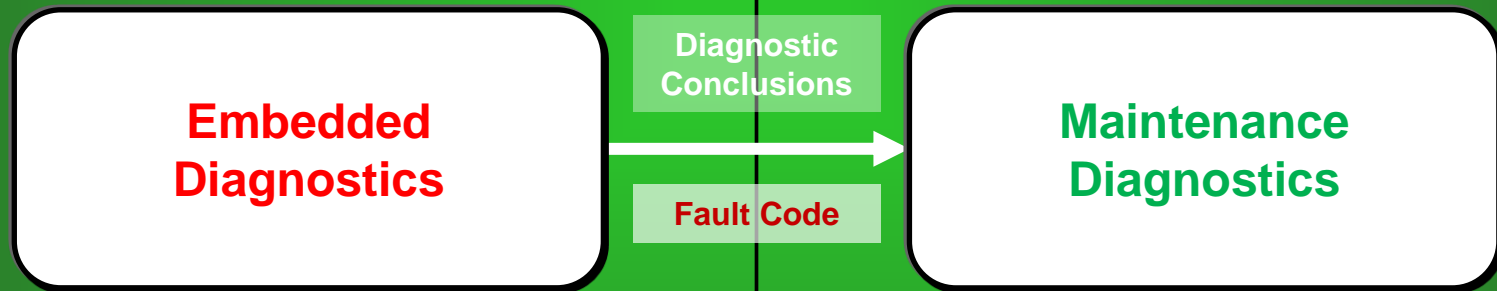
The *eXpress* Maintenance Module allows you to take the conclusions reached by one level of diagnostics and automatically utilize them within another diagnostic procedure. This is useful in a variety of situations, including:

- Integrating Embedded Diagnostics and Maintenance Diagnostics
- Integrating Diagnostics at Different Maintenance Facilities
- Integrating Acceptance Testing with Troubleshooting during Production
- Integrating Diagnostic Procedures on Automatic Test Equipment

# Bridging Multiple Levels of Diagnostics Using the *eXpress* Maintenance Module

On Board the System

Maintenance Facility



# Bridging Multiple Levels of Diagnostics Using the *eXpress* Maintenance Module

Diagnostics In the Field

Maintenance Depot

**O-Level  
(Operational)  
Diagnostics**

Diagnostic  
Conclusions

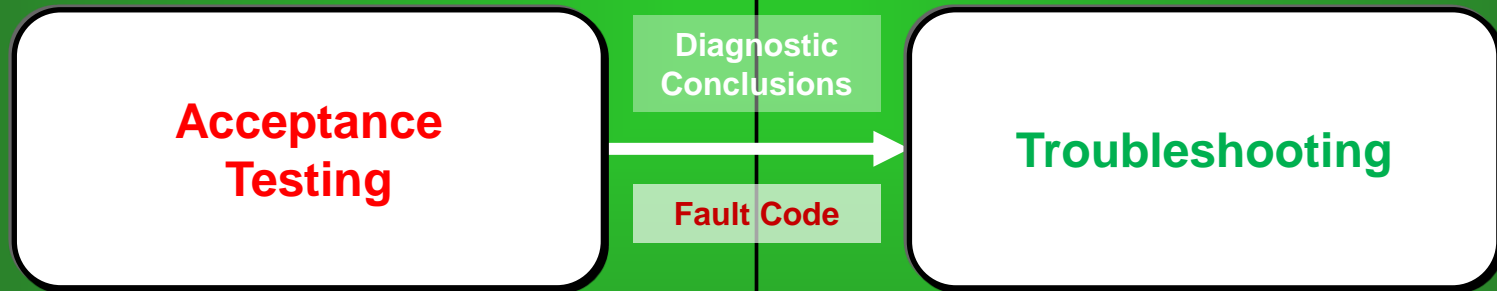
Fault Code

**I-Level  
(Intermediate)  
Diagnostics**

# Bridging Multiple Levels of Diagnostics Using the *eXpress* Maintenance Module

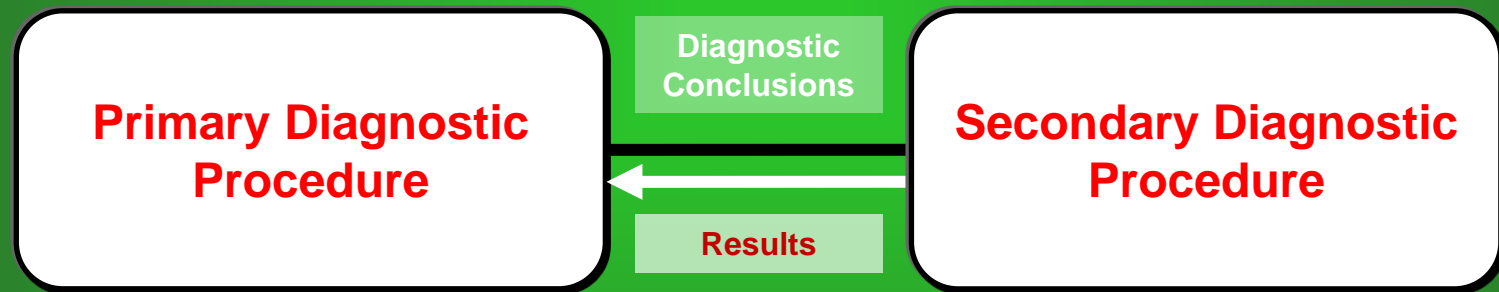
Production Floor

Troubleshooting Group



# Bridging Multiple Levels of Diagnostics Using the *eXpress* Maintenance Module

Automatic Test Equipment



# Fault Templates

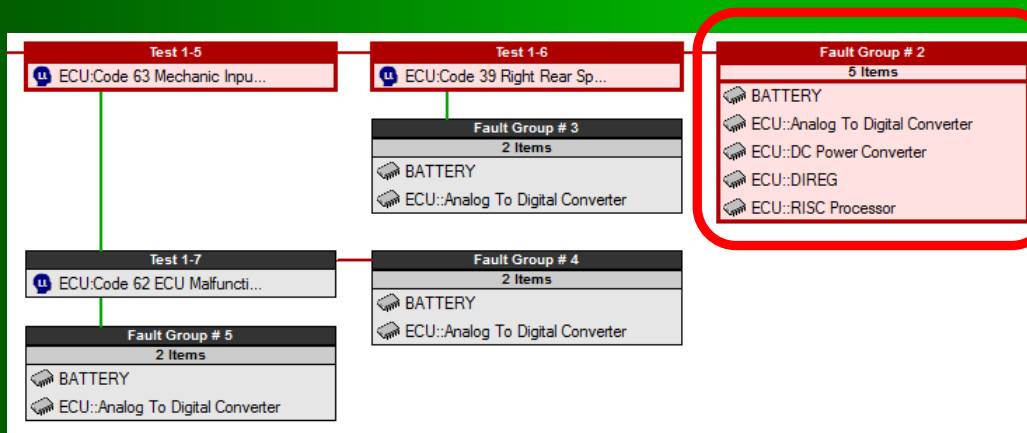
Many of the capabilities of the **eXpress** Maintenance Module are based on the creation of Fault Templates—representations within an **eXpress** design of the different conclusions possible during diagnostics. Here are some of the characteristics of Fault Templates in **eXpress**

- Can be derived from Fault Groups isolated within **eXpress**
- Can be defined in terms of Repair Items or Root Failures
- Can be assigned Attribute values (e.g. Fault Codes)
- Can be used as the basis for special Template-based Tests
- Can be compared against later diagnostics to determine status
- Can be used as a framework for developing Customized Repair Procedures (**Spring 2016**)

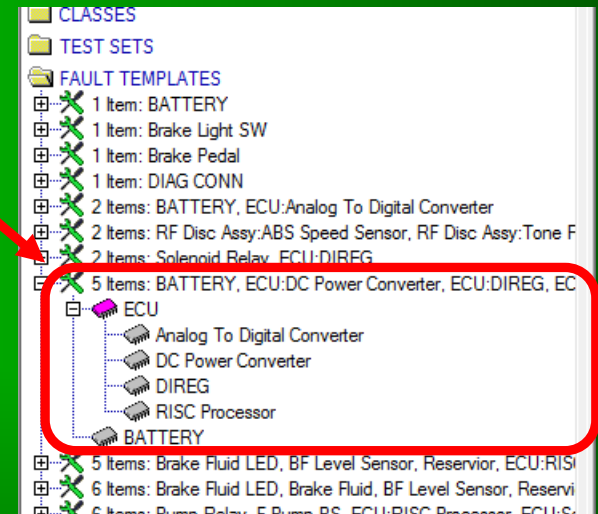


# Creating Fault Templates

After calculating diagnostics, use the “Create Fault Templates” operation to add a Fault Template to the *eXpress* design for each unique set of items (or root failures) that can be isolated by the diagnostics.



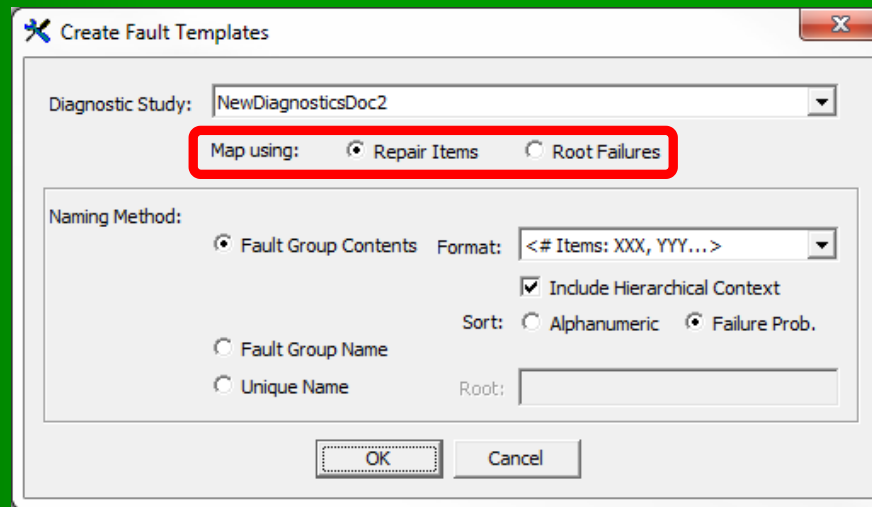
Excerpt from the Diagnostic Flow Diagram in an *eXpress* diagnostic study



Excerpt from the Explorer Tree, showing Fault Templates added to an *eXpress* design file

# Repair Items vs. Root Failures

The “Create Fault Templates” operation provides two different ways in which isolated fault groups can be mapped into Fault Templates:



Map using **Repair Items**

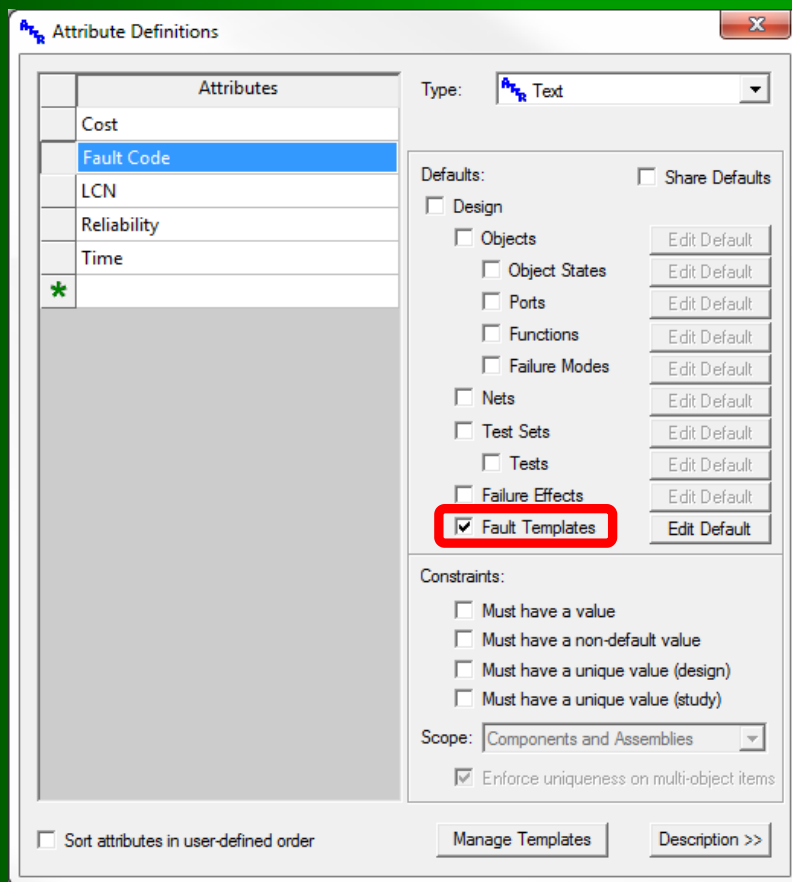
Fault Templates are created for each unique set of repair items. Because diagnostic conclusions are abstracted to the level of repair, this approach results in fewer Fault Templates.

Map using **Root Failures**

Fault Templates are created for each unique set of lowest-level failures. This approach results in more Fault Templates; however, templates capture the precise conclusions reached by the diagnostics.

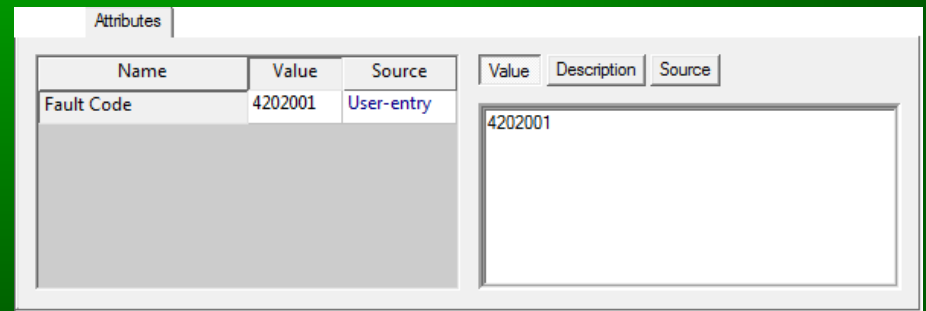
# Fault Template Attributes

Like many other elements in an *eXpress* model, attributes can be defined for Fault Templates.



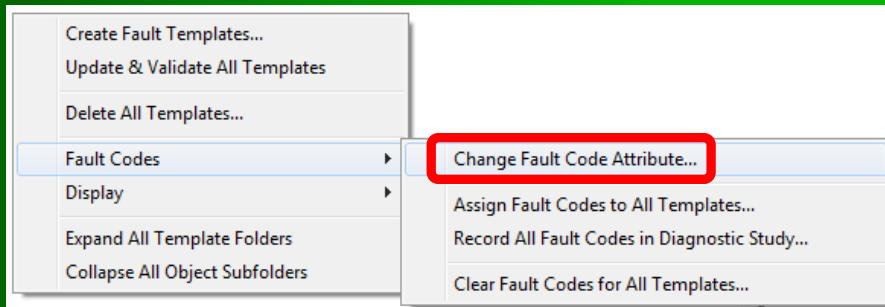
On the Attribute Definitions dialog depicted at left, an attribute named “Fault Code” has been defined for Fault Templates.

When you click on a fault template in the Explorer tree, the Attributes panel is displayed (see below). Use this panel to view or edit attribute values. Fault Template attributes can also be edited in Grid View.



# The Fault Code Attribute

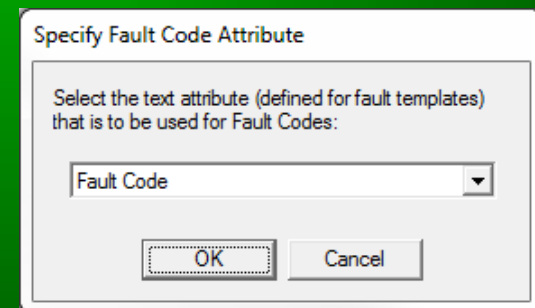
*eXpress* allows you to designate one text attribute associated with Fault Templates as being the Fault Code attribute (this attribute does not need to be named “Fault Code”). This attribute is then automatically handled in several areas of *eXpress*, including the following:



- Fault Codes Report
- Fault Signatures Report
- Fault Insertion Report
- Desktop Fault Insertion
- Explorer Tree

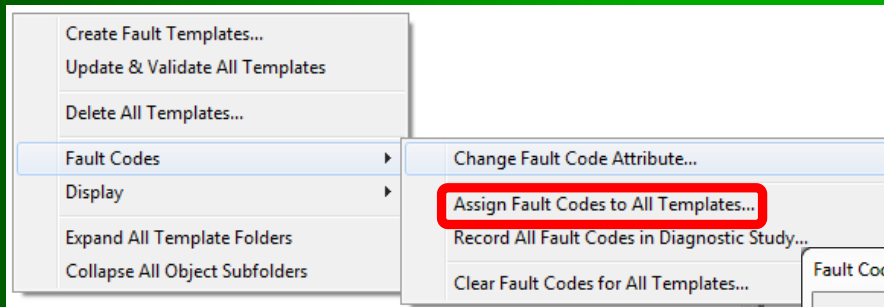
The Fault Templates menu (depicted above) has a section dedicated to the Fault Code attribute.

The dialog at right is used to specify (or change) which attribute is to be used as the Fault Code attribute.



# Automatic Fault Code Assignment

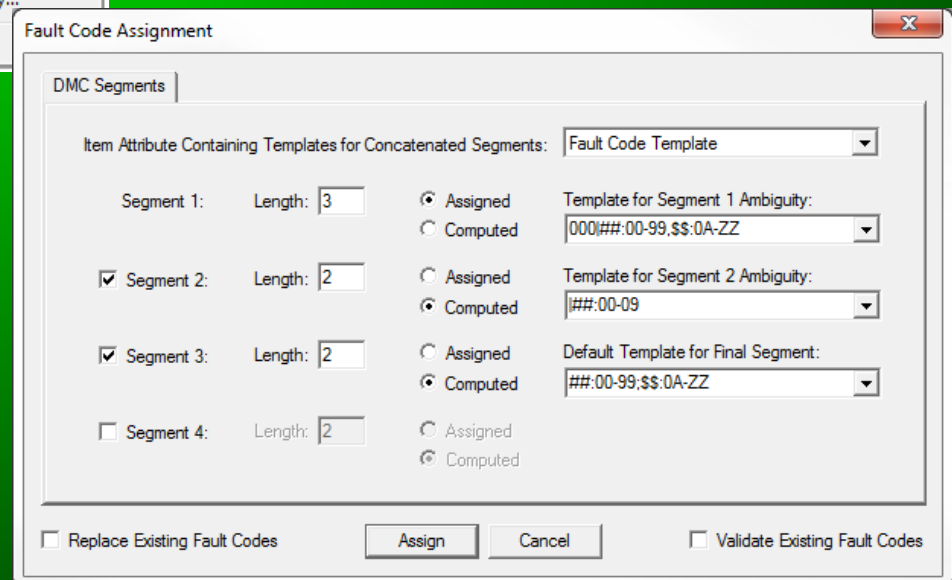
For large designs, there may be thousands of Fault Templates—each needing a unique fault code. To help with this, the Maintenance Module provides a Fault Code Assignment operation that constructs fault codes based on the contents of each Fault Template.



This feature is invoked using an operation listed in the Fault Codes section of the Fault Templates menu (depicted at left).

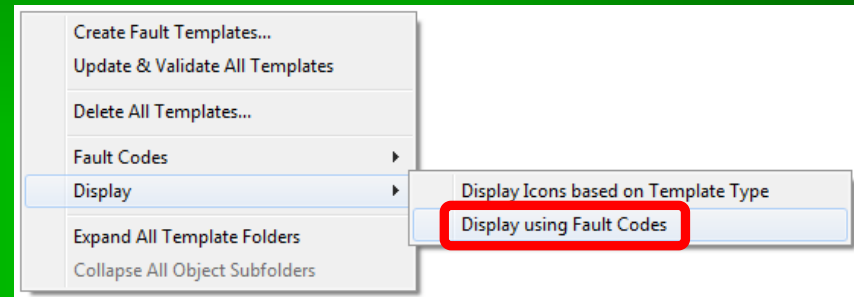
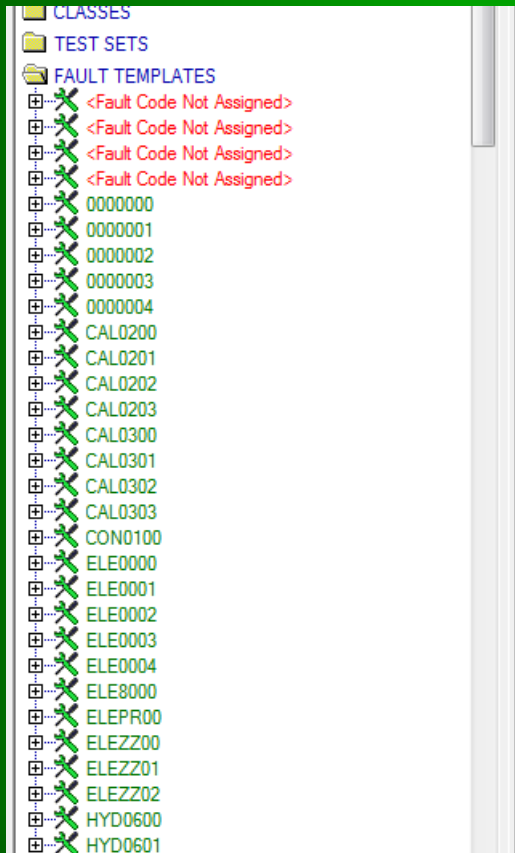
The Fault Code Assignment dialog (shown at right) is used to specify templates that define the content of the segments that are concatenated to create each fault code.

Segment templates are stored in object attributes, with special templates (on the dialog) for segments that must represent ambiguity in the associated Fault Template.



# Fault Codes in the Explorer Tree

Once fault codes have been assigned to Fault Templates, you can choose to have them displayed (rather than the Fault Template names) in the Explorer Tree.



In the Fault Templates menu (depicted above), there is a section that lets you control how Fault Templates are displayed in the Explorer Tree.

When Fault Templates are listed by fault code in the Explorer Tree (as shown in the example at left), any Fault Templates that have not been assigned a fault code are displayed in red.

# The Fault Codes Report

The Fault Codes Report provides a user-customizable listing of Fault Templates. This report is particularly useful when “handing off” information to other development efforts (such as the writing of repair procedures).

Fault Codes Report

Contents:

Fault Template Name  Number of Repair Items

Fault Template Status  Mapped Elements

Attributes for Mapped Objects (not available in RTF reports):

- Objects
  - General Properties
    - Description
  - Attributes
    - Cost
    - Fault Code Template
    - Item Type
    - LCN
    - Part Number
    - Reliability

Sort Order:

Sort by Fault Code

Filters:

by Status:  new  used  unused  invalid

by Code:   contains  starts with

by Size:  Include only if mapped to multiple objects

by Mapping:  (eg., xxx | yyy)

by Selection:  Include only if mapped to selected object(s)

58 Fault Codes meet the filtering criteria

Report Format:

Excel Spreadsheet

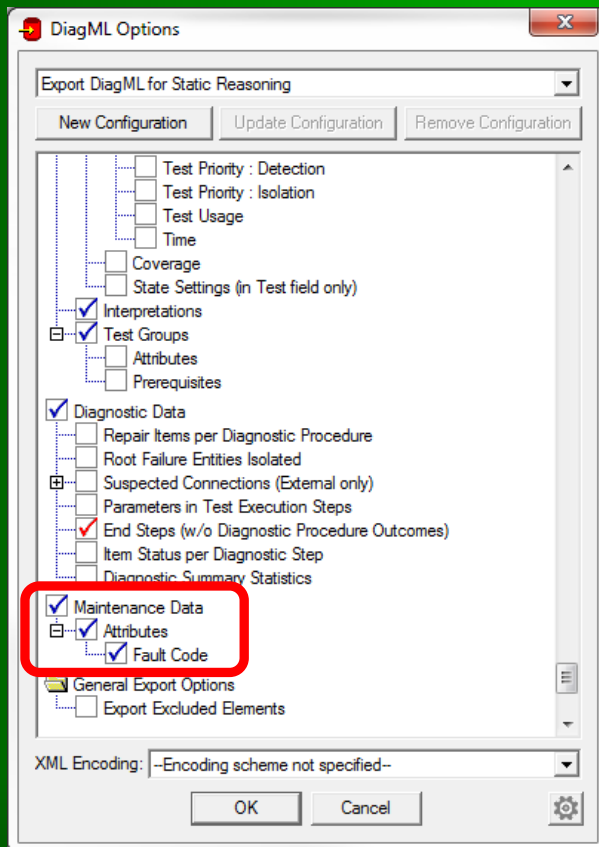
OK Cancel

The options dialog for the Fault Codes Report (shown at left), allows you to select which columns will appear in the report, as well as the order in which the data is to be sorted. There are a variety of filters so that the report contains only Fault Templates that are relevant to the task at hand. You can also choose whether the report is to be created in RTF, as a spreadsheet in MS Excel (shown below) or as an XML file.

Fault Codes Report			
##	Fault Code	## Items	Mapped Element(s)
1	000000	6	ECU::DIREG F Pump PS Pump Relay R Pump PS ECU::RISC Processor ECU::Solenoid Controller CMOS Switch
2	000001	2	LF Disc Assy::ABS Speed Sensor LF Disc Assy::Tone Ring
4	000002	2	RF Disc Assy::ABS Speed Sensor RF Disc Assy::Tone Ring
5	000003	2	ECU::DIREG SENS CONN 2A
8	CAL0200	1	RF Disc Assy::Caliper Assy
9	CAL0201	1	RR Disc Assy::Caliper Assy

# Exporting Fault Codes to DiagML

When exporting diagnostic data from *eXpress*, attributes can be included for each `<MaintenanceProcedure>` (isolated fault group) in the exported DiagML file. These attributes are taken from the Fault Templates in the *eXpress* model that correspond to each `<MaintenanceProcedure>` record.



```
<MaintenanceProcedure>
  <MaintenanceProcedure_ID>536871004</MaintenanceProcedure_ID>
  <Name>Fault Group # 91</Name>
  <Description>Replace 2 Items</Description>
  <MaintenanceAction>
    <Type>replace</Type>
    <DesignItem>18874411</DesignItem>
    <IsolatedFailProb>0.002639</IsolatedFailProb>
  </MaintenanceAction>
  <MaintenanceAction>
    <Type>replace</Type>
    <DesignItem>18874412</DesignItem>
    <IsolatedFailProb>0.000731</IsolatedFailProb>
  </MaintenanceAction>
  <MaintProc_Attribute>
    <Attribute_ID>Attr-Fault Code</Attribute_ID>
    <Name>Fault Code</Name>
    <Value>
      <PrimitiveValue>
        <string val="0000008"/>
      </PrimitiveValue>
    </Value>
  </MaintProc_Attribute>
</MaintenanceProcedure>
```

In this example, the DiagML export has been configured so that the attribute "Fault Code" is included for each `<MaintenanceProcedure>` (isolated fault group) in the exported DiagML file.



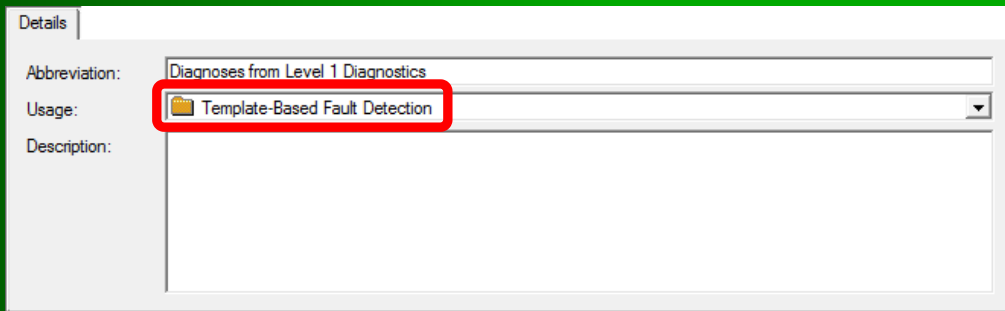
# Multiple Levels of Diagnostics

It is sometimes desirable to create multiple levels of diagnostics within *eXpress* (for instance, first-level diagnostics representing the embedded capability of a system and second-level diagnostics that represents the testing performed in a maintenance facility). The *eXpress* Maintenance Module provides you with the special operations that are needed to perform this process:

- Calculate the first-level diagnostics in *eXpress*
- Create Fault Templates representing the isolated fault groups
- Assign fault codes to the Fault Templates
- Define special tests based on the Fault Templates
- Generate second-level diagnostics using template-based detection

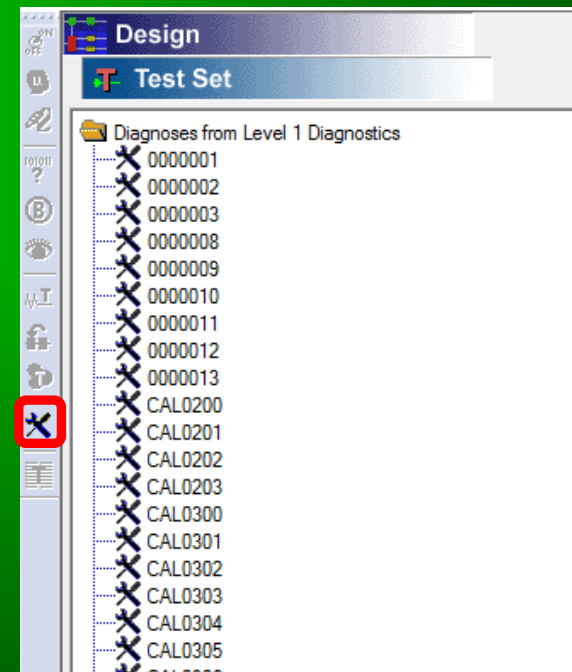
# Tests Based on Fault Templates

When more than one level of diagnostics are to be developed in *eXpress*, Fault Templates based on the isolated fault groups in the first level of diagnostics can be used to create special tests to be used as entry points into the second level of diagnostics.



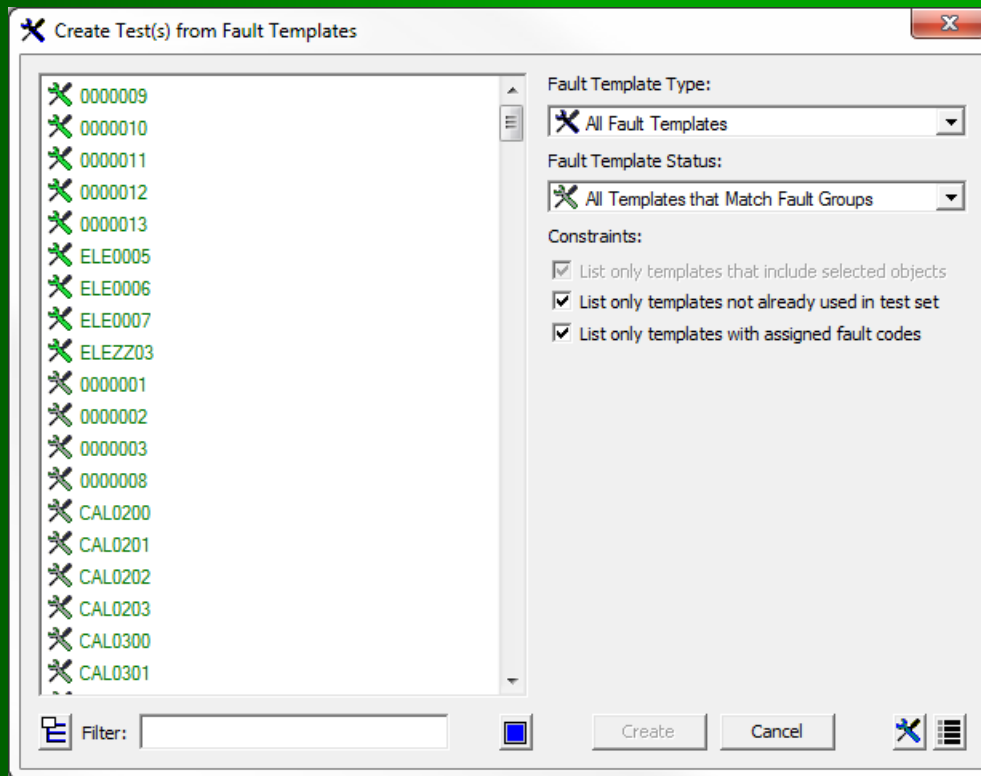
Before you can create tests based on Fault Templates, you must first create a test set and change its usage setting to “Template-Based Fault Detection” on the test set Details panel (as shown above).

When you edit this test set, all of the usual icons for creating tests will be disabled; instead, a special “New Test(s) from Fault Templates” icon will be enabled (see example at right). Click on this icon to create tests based on the Fault Templates in the model.



# Creating Template-Based Tests

Template-based tests can be created for selected Fault Templates. The coverage of each test will be equivalent to the contents (repair items or root failures) of the corresponding Fault Template.

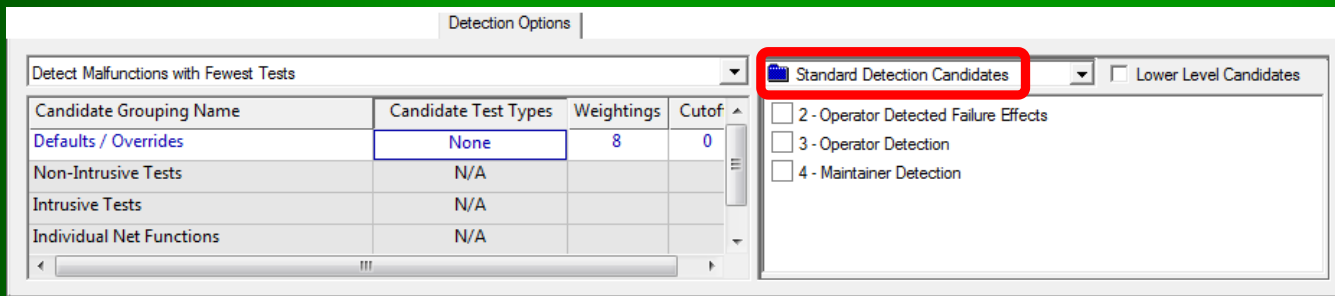


Simply select the desired Fault Templates from the list and then click on the “Create” button. When fault codes have been assigned, the Fault Templates will be listed by fault code. To display the Fault Template names instead, simply click on the icon in the lower-right corner of the dialog.

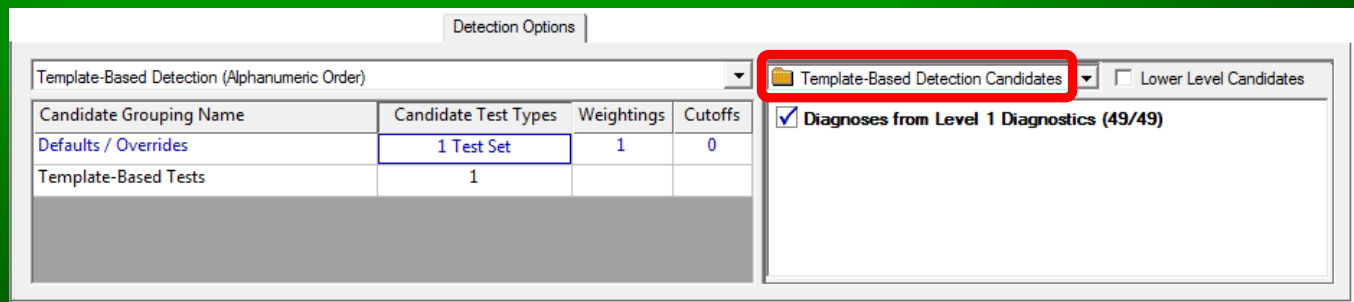
This dialog also has a number of filters and constraints to facilitate the quick selection of the desired Fault Templates.

# Generating Second-Level Diagnostics

In *eXpress*, diagnostics can be created using template-based detection tests (tests that represent knowledge from the previous diagnostic level).

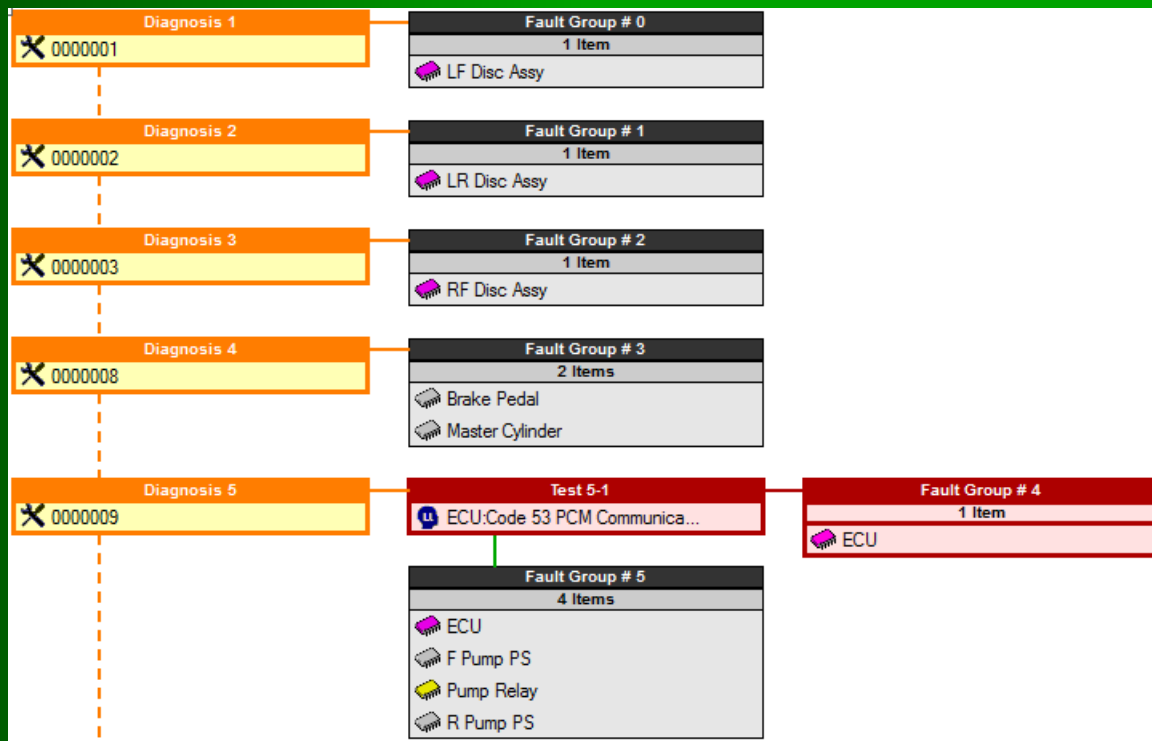


When calculating first-level diagnostics in *eXpress*, the candidate test sets for fault detection are selected on the Detection Options panel (depicted above). To select the candidate test sets to be used when calculating second-level diagnostics, you must select “Template-Based Detection Candidates” in the dropbox on the right-hand side of the panel (as shown in the example below).



# Fault Codes & Multi-Level Diagnostics

When implemented for a fielded system, second-level diagnostics begin where the previous level left off, with the assigned fault codes providing the links between the different levels of diagnostics.



In the Diagnostic Flow Diagram, template-based detection tests, when highlighted, are colored orange. These tests are labeled as diagnoses (rather than tests), since they represent outcomes of the previous level diagnostics.

Note that the fault code is used as the name of each test node. When implemented in the field, fault codes produced by the first-level diagnostics provide entry points into the second-level diagnostics—encapsulating, in effect, all previous diagnostic knowledge within the fault code.

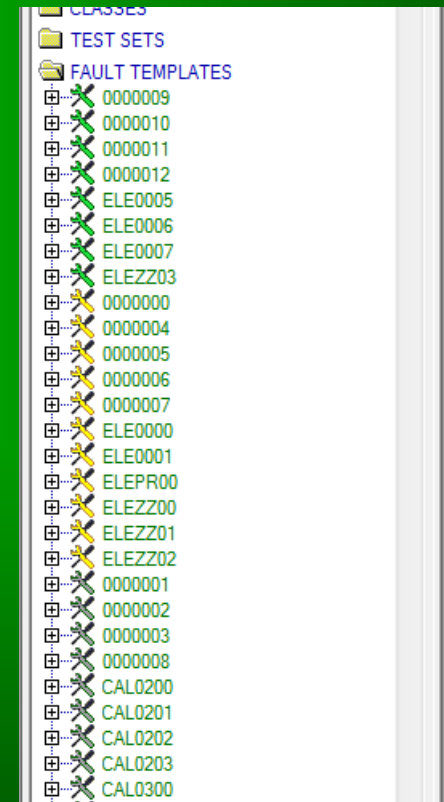
# Support for Concurrent Engineering

In *eXpress*, Fault Templates are categorized by usage. When you update templates using a new version of the diagnostics, the usage settings are updated as well. There are four usage categories—each with a different color icon: New (green), Used (gray), Unused (yellow) and Invalid (red).

In the Explorer Tree, Fault Templates are sorted first by usage, then by name (or fault code). This makes it easy to identify which templates are new and which are obsolete.

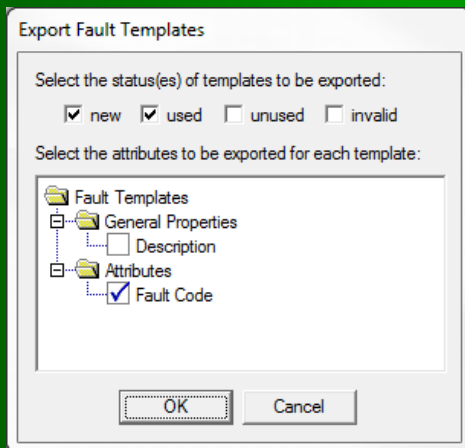
In the example at right, newly-added Fault Templates appear at the top of the list. Next are obsolete Fault Templates—templates that no longer correspond to a fault group in the latest run of the diagnostics. Listed after these are previously-created templates that still map to fault groups in the latest diagnostics.

Several features allow you to filter templates by usage. For instance, you can create a Fault Codes Report containing only new fault codes ...or only obsolete fault codes. This capability has been designed to facilitate concurrent development efforts. Even when you are working with thousands of fault codes, it is easy to identify what has changed.



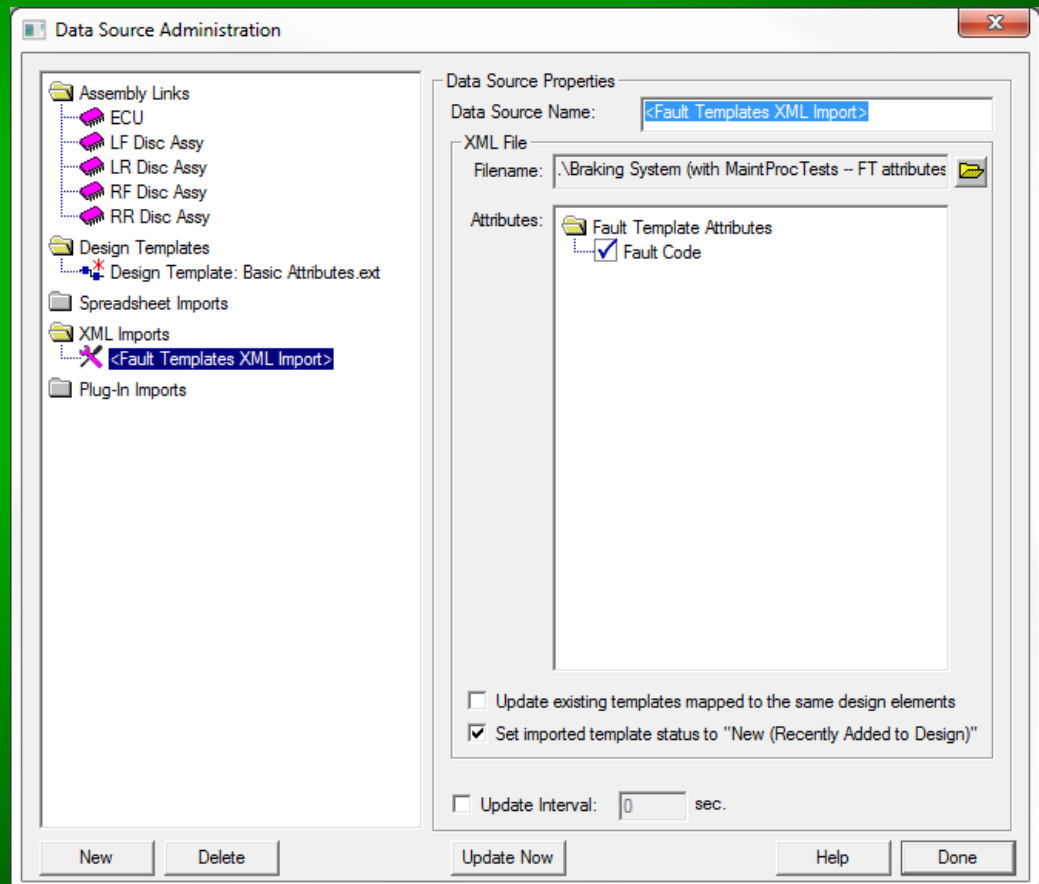
# Exporting & Importing Fault Templates

Fault Templates can be moved between models that represent different versions of a system using the Fault Template Export & Import operations.



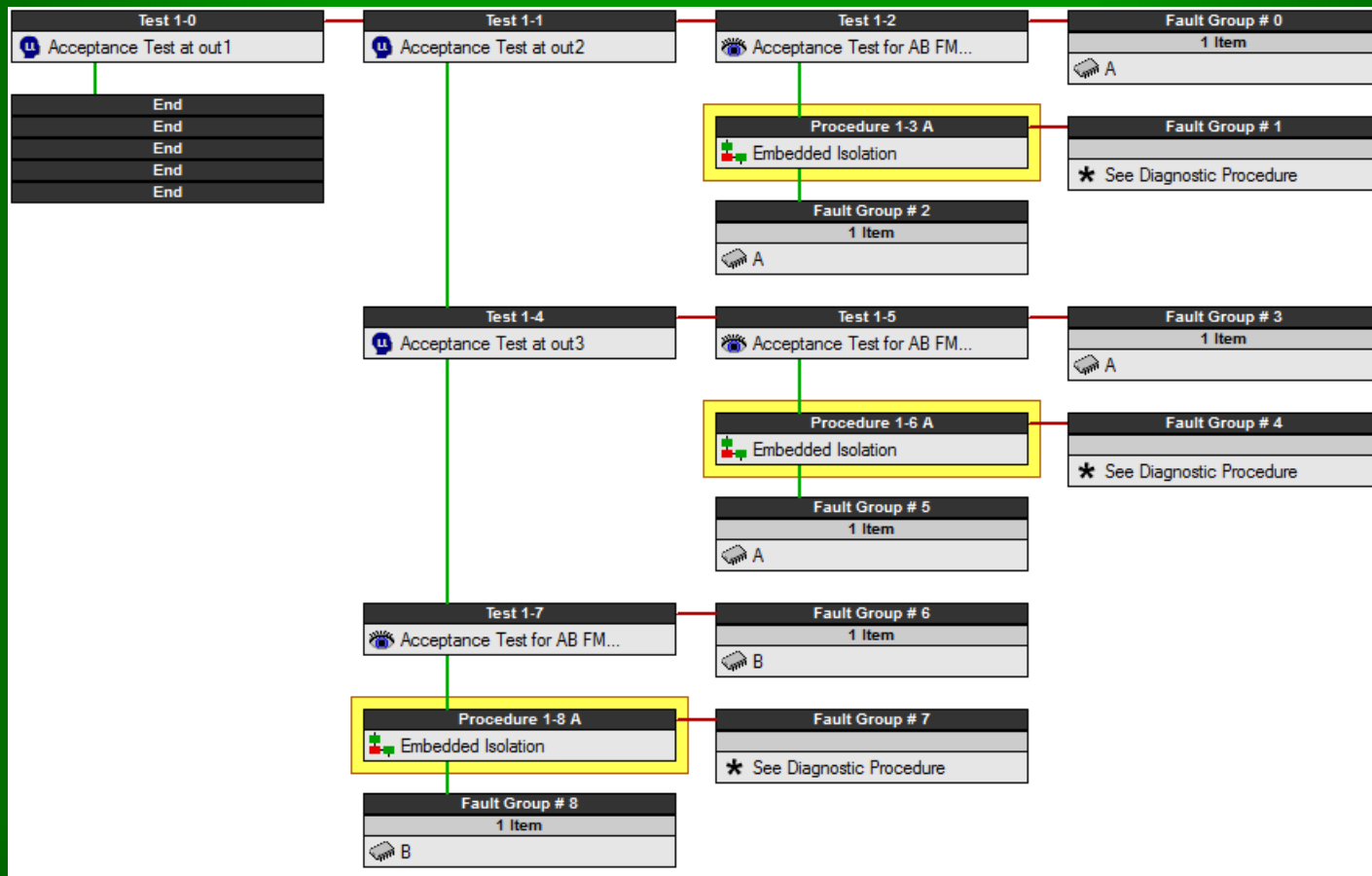
The Export Fault Templates dialog (depicted above) can be invoked using an operation in the Export section of the main File menu.

Fault Templates are imported using an XML import on the Data Source Administration dialog (see right).



# Segmented Diagnostics

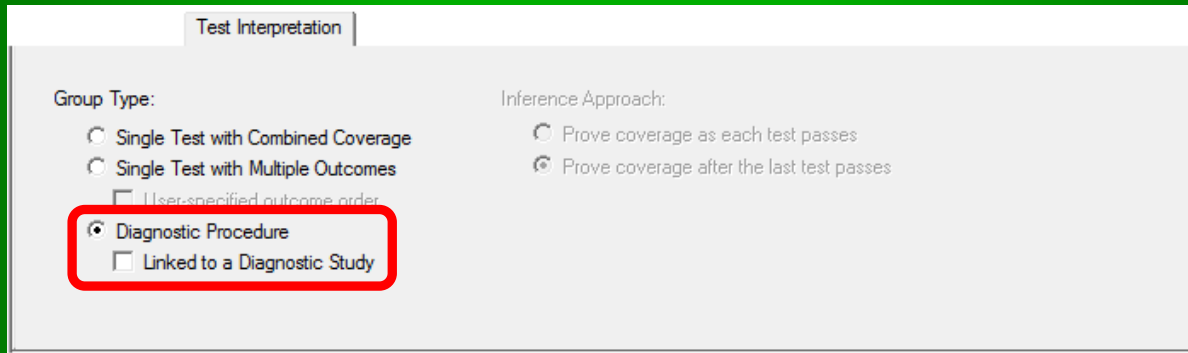
The *eXpress* Maintenance Module allows you to create Segmented Diagnostics—diagnostic sequences that call other diagnostic sequences.





# Diagnostic Procedure Tests

Segmented Diagnostics are created using “Diagnostic Procedure” tests, special group tests (available only when the **eXpress** Maintenance Module is licensed) that represent secondary diagnostic procedures called from within a primary procedure.

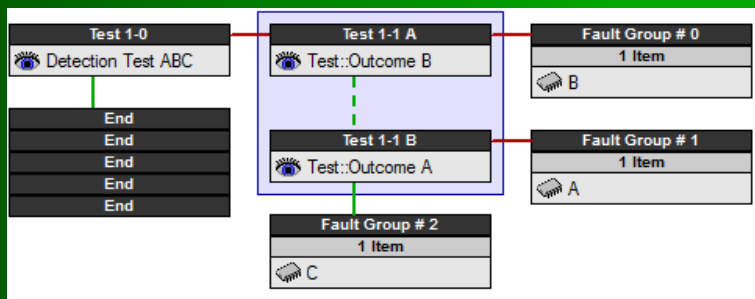


The screenshot shows a software interface titled "Test Interpretation". It contains two main sections: "Group Type:" and "Inference Approach:". Under "Group Type:", there are four radio button options: "Single Test with Combined Coverage", "Single Test with Multiple Outcomes", "Diagnostic Procedure" (which is selected and highlighted with a red rectangle), and "Linked to a Diagnostic Study". There is also a checkbox option "User-specified outcome order" which is unchecked. Under "Inference Approach:", there are two radio button options: "Prove coverage as each test passes" and "Prove coverage after the last test passes", with the latter being selected.

A Diagnostic Procedure test is a group test whose type has been set to “Diagnostic Procedure” on the Test Interpretation panel (shown above). The individual tests within the group should represent all the possible outcomes of the diagnostic procedure. Diagnostic procedure tests can be optionally linked to a diagnostic study that represents that procedure.

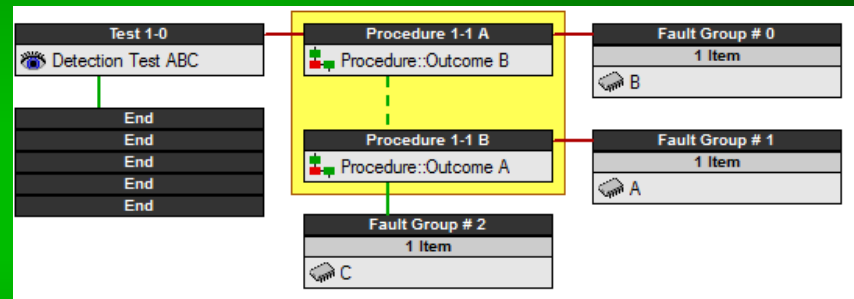
# Diagnostic Procedure Tests vs. Multiple-Outcome Tests

Diagnostic Procedure tests are very similar to multiple-outcome tests. Other than cosmetic differences (how they appear on the screen), there are three primary differences between the two types of group tests:



## Multiple-Outcome Test

1. The test's outcomes are assumed to be mutually exclusive
2. The test is exported to DiagML as one test in a single diagnostic procedure.
3. Subsequent isolation tests are not impacted

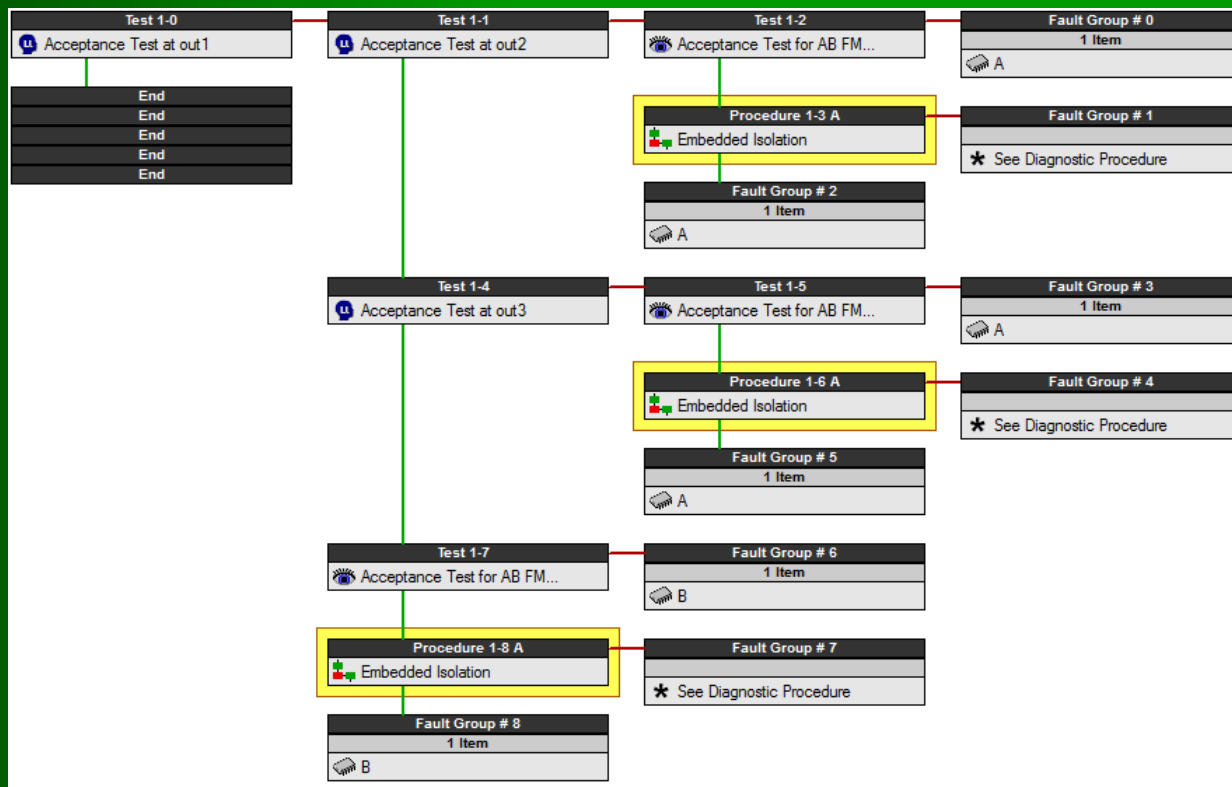


## Diagnostic Procedure Test

1. Each procedure may return multiple outcomes (handled in **DSI Workbench**)
2. The test is exported to DiagML as a separate diagnostic procedure
3. Subsequent isolation tests are exported as part of the secondary diagnostic procedure

# Linked Diagnostic Procedures

“Linked” Diagnostic Procedure tests are used to represent diagnostics that have been developed in a separate diagnostic study.



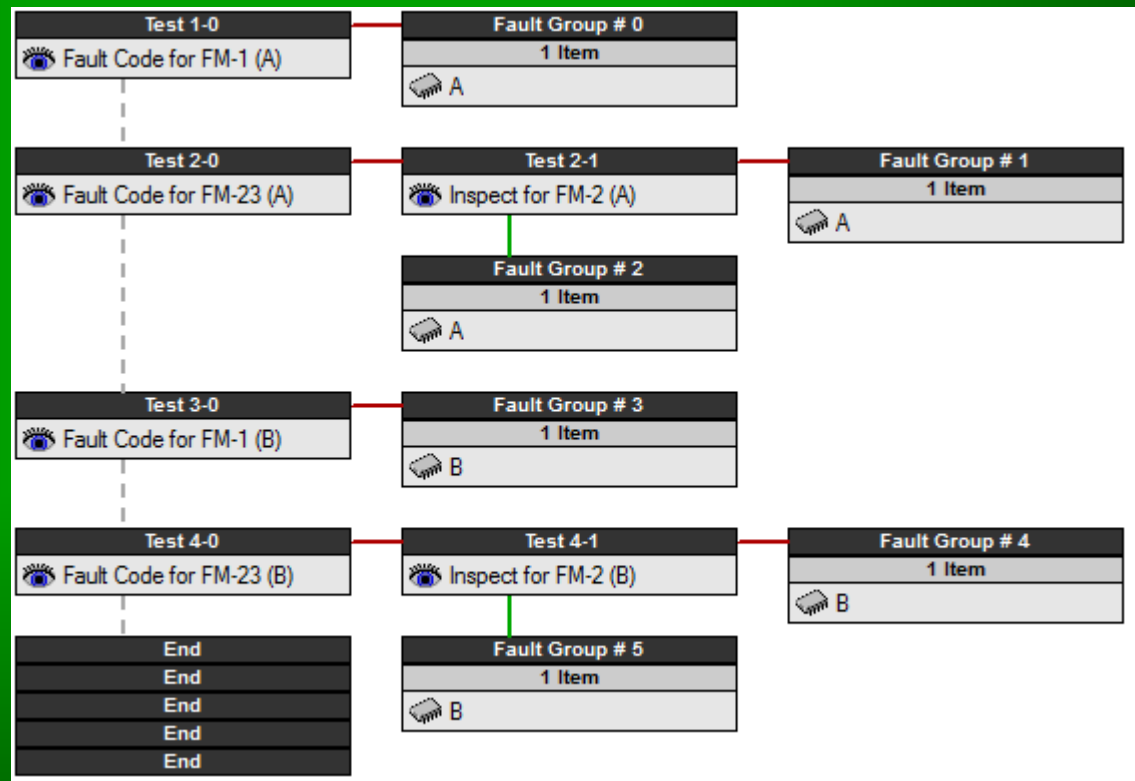
For Linked Diagnostic Procedures, individual outcomes are not included in the primary tree. In the Diagnostic Flow Diagram, a special fault group cell appears where one of these procedures fails, indicating that diagnostics are continued within a secondary procedure.

# Linked Diagnostic Procedures: The Secondary Diagnostic Procedure

For Linked Diagnostic Procedures, secondary diagnostics must be calculated within a separate diagnostic study.

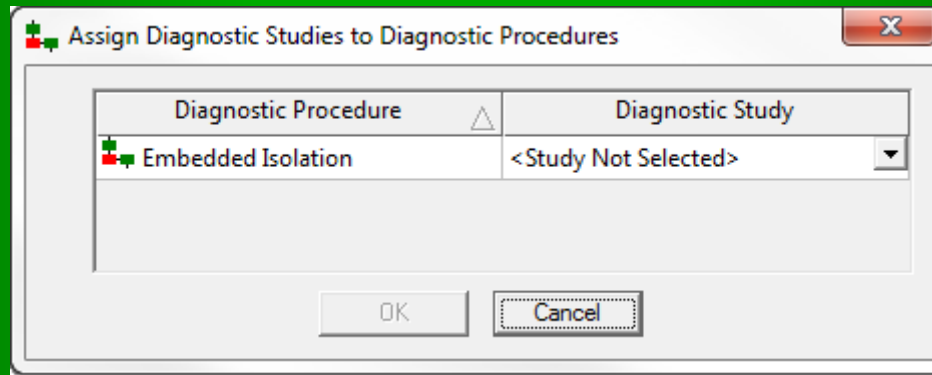
The secondary diagnostics should be generated using “Independent Mode”, with the individual tests that were grouped together in the Diagnostic Procedure test now used individually as detection tests.

For each detection test (that is, each outcome of the diagnostic procedure), additional troubleshooting tests can be added to complete fault isolation.



# Linked Diagnostic Procedures: Exporting to DiagML

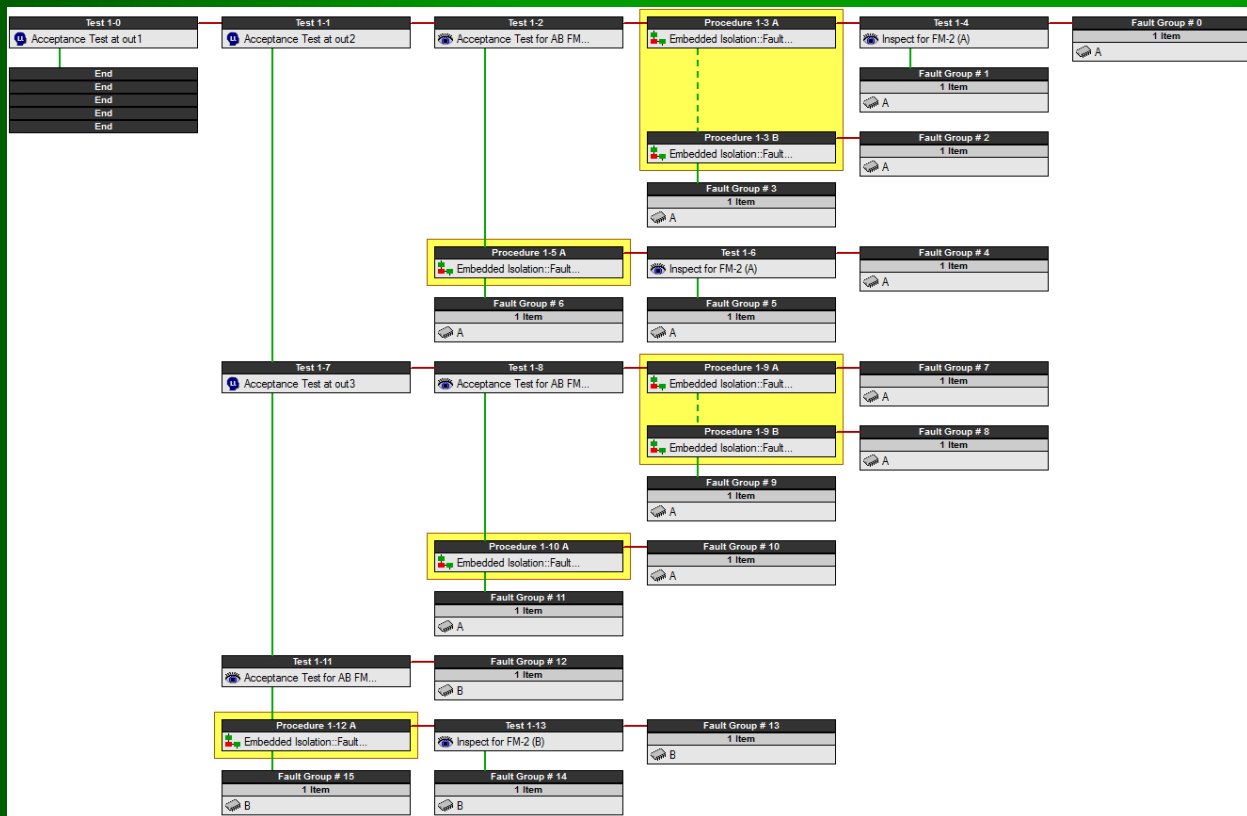
When exporting Linked Diagnostic Procedures to DiagML, you must first open all primary and secondary diagnostic studies within *eXpress*. During the DiagML export, the following dialog will appear:



For each Diagnostic Procedure, select the diagnostic study that contains the associated secondary diagnostic sequence. In the exported DiagML file, a single <DiagnosticProcedure> section will be created for each Diagnostic Procedure test (with all instances of the test pointing to that procedure).

# Unlinked Diagnostic Procedures

For Unlinked Diagnostic Procedure tests, secondary diagnostics are developed within the primary diagnostic study.



For Unlinked Diagnostic Procedures, individual outcomes are included in the primary tree. In the Diagnostic Flow Diagram, subsequent fault isolation appears to the right of each procedure outcome.

When exported to DiagML, a <DiagnosticProcedure> section will be created for each instance of the Diagnostic Procedure test (since all outcomes may not apply to each instance)

# Customized / Optimized Repair Procedures (Spring 2016)

The **eXpress** Fault Isolation and Subset FD/FI reports provide Fault Isolation metrics (based on block replacement) and Fault Resolution metrics (based on serial replacement). For each fault group, the serial replacement order is determined by failure probability (“Lambda Search”). The Maintenance Module adds other possibilities, including:

- Serial Replacement based on User-Defined Criteria (Failure Probability, Replacement Cost, Repair Time, etc.)
- Alternative Maintenance Actions (Adjust, Repair, Replace, etc.)
- Serial Replacement at Multiple Levels of Indenture
- Additional (“Opportunistic”) Replacements

Customized & Optimized Repair Procedures will not only impact metrics calculated in **eXpress**, but also simulations in **STAGE** and the ranking of suspected Repair Items in **DSI Workbench**.