

---

Author(s)	Martin Huck
Restrictions	Public Document
Abstract	This application note describes how traceability between DOORS and CANoe Test Modules can be established. Furthermore it describes how to generate XML test modules from DOORS modules and vice versa.

---

### **Table of Contents**

1	Overview .....	1
2	DOORS Introduction .....	1
3	Test Feature Set with DOORS Support.....	4
3.1	Traceability via Test Case Identifiers and External References .....	4
3.2	CANoe Test Feature Set Extension .....	5
4	Use Cases .....	7
4.1	Test Feature Set DOORS Extension Installation .....	7
4.2	Module Generation.....	8
4.2.1	XML Test Module Creation .....	8
4.2.2	TDM Creation.....	10
4.3	Navigation .....	12
4.3.1	Navigation to the associated XML Test Module .....	12
4.3.2	Navigation from the Test Report to DOORS .....	13
4.4	Module Updates .....	13
4.4.1	XML Test Module Update .....	13
4.4.2	DOORS Module Update .....	14
4.5	Adding Traceability to an existing DOORS Database.....	16
5	Additional Resources .....	17
6	Glossary .....	18
7	Contacts .....	18

## **1 Overview**

In many projects DOORS (trademark of Telelogic AB) is e.g. used to collect, structure and manage requirements. In the same way DOORS can be used to manage test case descriptions and they can be linked to the corresponding requirements.

This application note describes how Vector CANoe supports traceability between external Test Modules and DOORS. Furthermore it demonstrates how the Vector CANoe Test Feature Set provides an enhanced and integrated approach to generate XML Test Modules from DOORS Modules and to keep them synchronized.

## **2 DOORS Introduction**

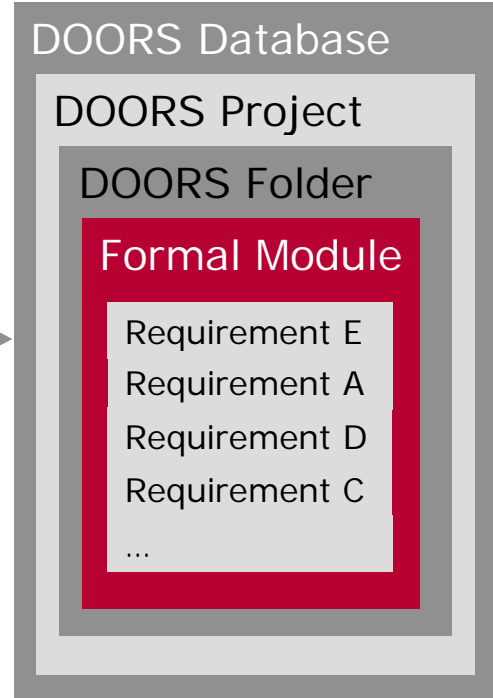
Usually project goals are defined as a set of initial requirements in an arbitrary formal representation, e.g. tables or documents. Requirements engineering will break down and structure these initial requirements. All requirements are inserted into DOORS Formal Modules Figure 1. From now on DOORS is used to maintain and further manage project requirements.

Several projects with their Formal Modules can be grouped into the same DOORS database. DOORS folders can be used to arrange Formal Modules hierarchically.

Initial documents with project requirements:

Feature	Release	Priority
Xxxx xx xxx xxx xxx x x Xxx xx.	1	Low
Xxx xxx xx x xxx xxx Xxxx Xx.	1	high
Xxxxx xxx xxx x xx X xx xxx.	1	medium
Xxxx xxx xxx xxx.	1	high
X x xxxx X x xxx xxxxx.	1	High
Xxxx xxxxx xxx xxx x xx xxx.	1	low

Requirements Engineering



```

1 Xxxx xx xxx xxx
xxx x x Xxx xx. Xxx xxx xx x xxx xxx Xx. Xxxxx xxx xxx x xx X xx xxx. X x
xxxx X x xxx xxxxxx. Xxx xxx xx x xxx xxx Xxxx Xx. Xxxxx xxx xxx x xx X xx xxx.
X x xxxx X x xxx xxxxxx. Xxx xxx xx x xxx xxx Xxxx Xx. Xxxxx xxx xxx x xx X xx
xxx. X x xxxxx X x xxx xxxxxx.
• Xxxx xxxxxx xxx xxx x xx xxx. Xxxx xxx xxx xxx.
• Xxxx xxxxxx xxx xxx x xx xxx. Xxxx xxx xxx xxx.
• Xxxx xxx xxx xxx.

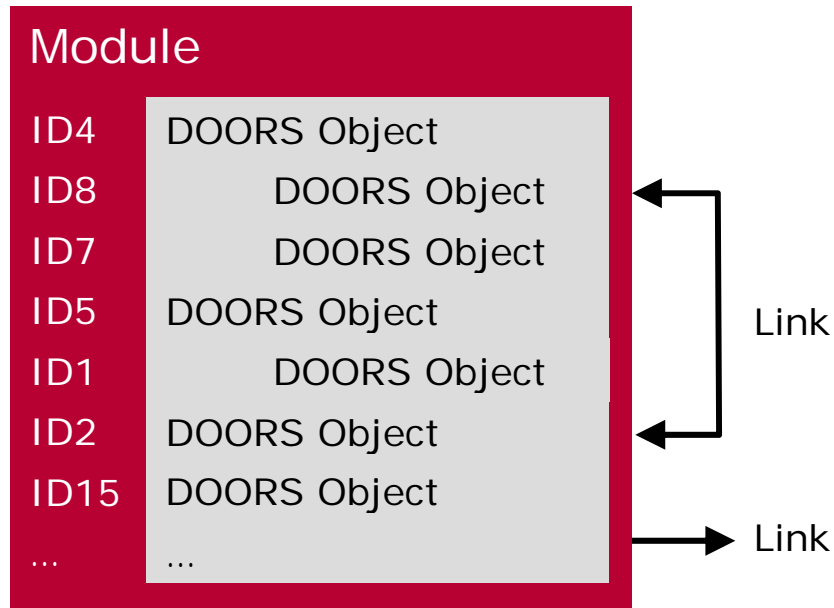
2 Xxxx xx xxx x xxx xxxxx x xxx
Xxx xxx xx x xxx xxx Xxxx Xx. Xxxxx xxx xxx x xx X xx xxx. X x xxxxx X x xxx
xxxxx. Xxx xxx xx x xxx xxx Xxxx Xx. Xxxxx xxx xxx x xx X xx xxx. X x xxxxx X x
xxx xxxxxx.
    
```

Figure 1: Requirements Engineering Process and DOORS

Within the Formal Module all requirements are stored in DOORS Objects. Every DOORS Object consists of an Object heading with heading number and a Text Object. Additional attributes with user defined content like 'Priority' or 'Release' can be inserted.

All objects are hierarchically arranged and get unique identifiers, see Figure 2. Once the identifier is assigned it cannot be modified or reused for another object. The identifier is a combination of an automatically generated number and a user definable prefix (in the example 'ID' is chosen). The prefix is configured when the Formal Module is created.

A DOORS object can be linked to one or more DOORS objects that are located in the same or another Formal Module and database. Links can also point to DOORS external destinations.



**Figure 2: Hierarchically arranged DOORS objects**

Figure 3 shows an example for the overall DOORS hierarchy and related screen shots of the main DOORS windows.

**Figure 3: DOORS Hierarchy and main windows for an example Database**

Beyond managing requirements Formal Modules are useful to organize test descriptions that are stored in DOORS Objects. These Formal Modules are independent from other Formal Modules with requirements. In particular there can be more or less Formal Modules and they can have any arbitrary hierarchy.

Very useful are links that connect test descriptions to their related requirements in other Formal Modules. These links allow navigation from requirements to test cases and vice versa.

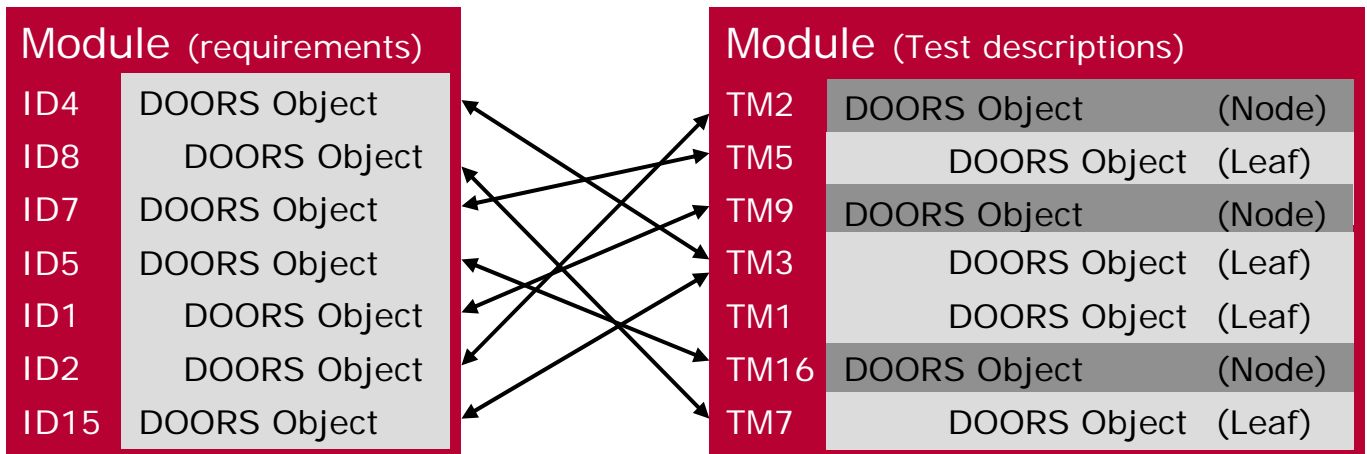


Figure 4: Linked DOORS requirements and test descriptions

In this application note Formal Modules that contain requirements are called **RQM** (Requirement Module) and those with test descriptions **TDM** (Test Description Module).

For DOORS TDM we further differentiate between

- **Leaf Objects** that are at the lowest hierarchy level and that do not have subordinate objects (e.g. TM5). In the following figures they have light grey as background colour.
- **Node Objects** that have subordinate objects (e.g. TM2). They have a grey background colour.

Please note that Leaf Object, Node Object, RQM and TDM aren't terms defined in DOORS. They are only introduced for convenience in this application note.

Beyond the described DOORS basics there are many more features and ways to use DOORS which are less important for understanding how the Test Feature Set can be integrated in DOORS.

### 3 Test Feature Set with DOORS Support

The next consequent step in structuring and managing project relevant information is to provide traceability from Test Modules as well as from Test Reports to DOORS.

Traceability can be already achieved with CANoe when inserting Test Case Identifiers and external references manually. A more sophisticated solution is provided with the Test Feature Set DOORS Extension and the Test Automation Editor.

#### 3.1 Traceability via Test Case Identifiers and External References

CAPL and XML Test Modules can define URL links to TDMs that are inserted into the Test Report. By clicking on the link in the report the TDM is opened and navigation from the Test Report to the test description is achieved.

In XML Test Modules this is done by adding an “externalref”, e.g.

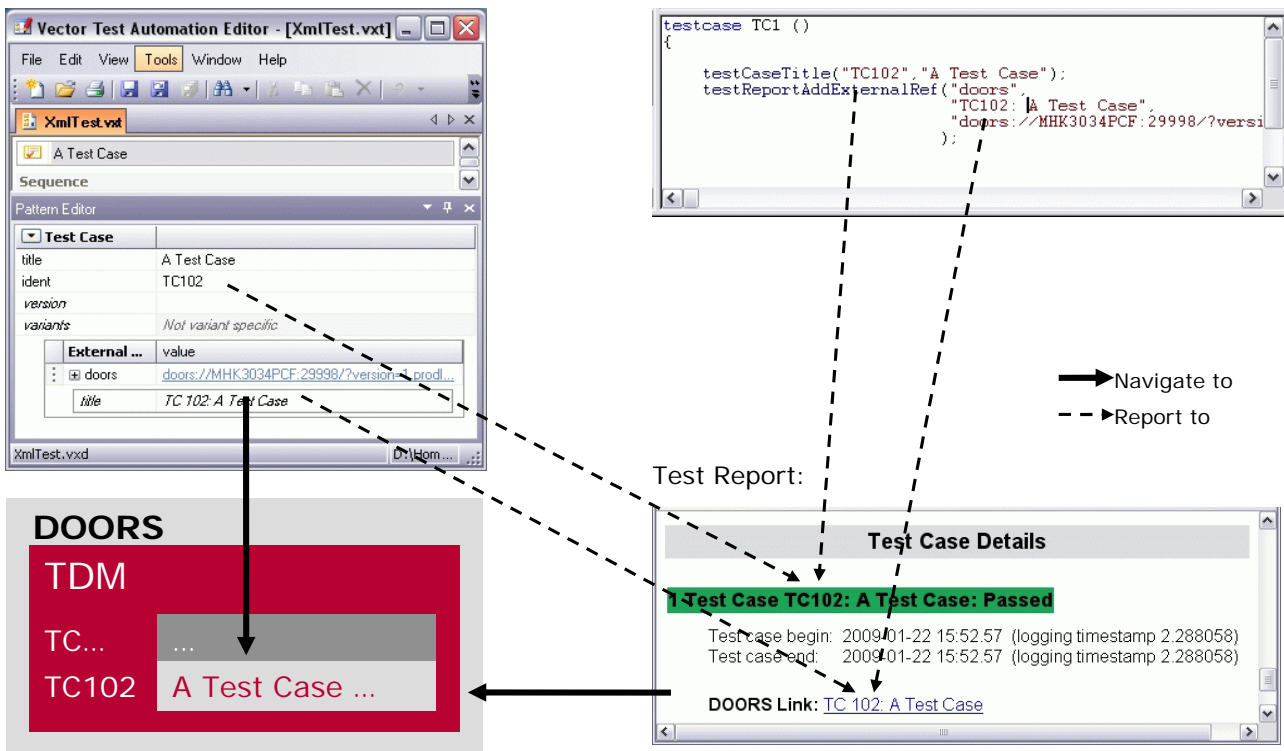
```
<testcase title="A Test Case" ident="TC102">
  <externalref type="doors" title="TC102: A Test Case">doors://... </externalref>
  ...
</testcase>
```

The ‘ident’ attribute of the Test Case in the Test Module is intended to be set to the identifier of the related DOORS object.

With the Test Automation Editor this can be done more conveniently, as shown in the upper left corner of Figure 5.

In CAPL Test Cases the commands “TestCaseTitle” and “TestReportAddExternalRef” are used to create links in the report. An example can be found in Figure 5.

**Tip:** The DOORS URL can be obtained in the DOORS Module Window by opening the context menu for a DOORS Object and selecting ‘Copy URL’. Afterwards it can be pasted into the CAPL code or XML description.



**Figure 5: Traceability with Test Case Identifiers and navigation with external references**

### 3.2 CANoe Test Feature Set Extension

The CANoe Test Feature Set DOORS extension adds a new menu to the DOORS Module Window that allows to

- navigate from a DOORS Object to the related XML Test Module, Test Group or Test Case
- generate XML Test Modules from TDMs
- generate TDMs from existing XML Test Modules
- synchronize TDMs and XML Test Modules when their counterpart has been changed.

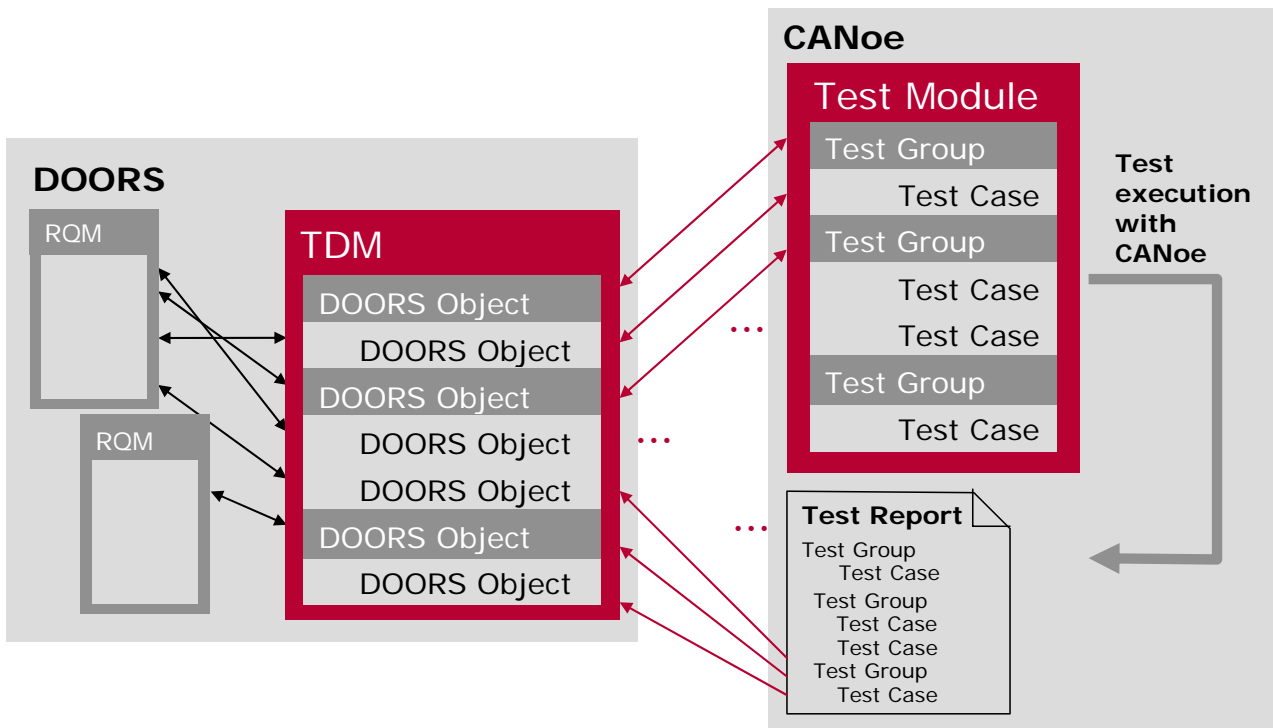
The TDM and the XML Test Module are associated and their individual objects are linked. The association is important when navigating to the XML Test Module or when updating it.

The TDM structure is one-to-one translated to the XML Test Module:

- TDM Node Objects result in XML Test Groups and their hierarchy level is kept.
- TDM Leaf Objects result in Test Cases

DOORS Descriptions and identifiers are inserted in the Test Groups respective Test Cases. After generation the XML Test Modules can be completed, i.e. test patterns are inserted and parameterized.

Also the test report that will be written during test execution contains the links from Test Groups and Test Cases to the related DOORS Objects.



**Figure 6: Relationship between TDM and XML Test Module**

Traceability allows answering e.g. the following questions:

- Which test cases are affected when a requirement changes?
- Which requirements are violated when a test case fails?
- Are all Test Cases implemented?
- Which requirements are covered by a test?

## 4 Use Cases

The following chapters demonstrate the usage of the CANoe Test Feature Set DOORS extension.

- The first use case shows how to install the new feature.
- Several detailed scenarios provide information about XML and TDM generation and navigation.
- Finally it is shown how TDMs and XML Test Modules can be kept up-to-date and what can be done when traceability has to be integrated into an existing DOORS database.

All scenarios contain screenshots of DOORS version 8.1 and reflect the features set of the Vector Test Automation DOORS Extension which is delivered with the Test Automation Editor 1.1.

### 4.1 Test Feature Set DOORS Extension Installation

#### Target:

- Extend DOORS with the Vector Test Automation menu.

#### Steps:

- On a PC with an installed DOORS client it is sufficient to install a new Test Automation Editor. When DOORS is started for the next time a new menu entry is present in the DOORS Module Window, see Figure 7.

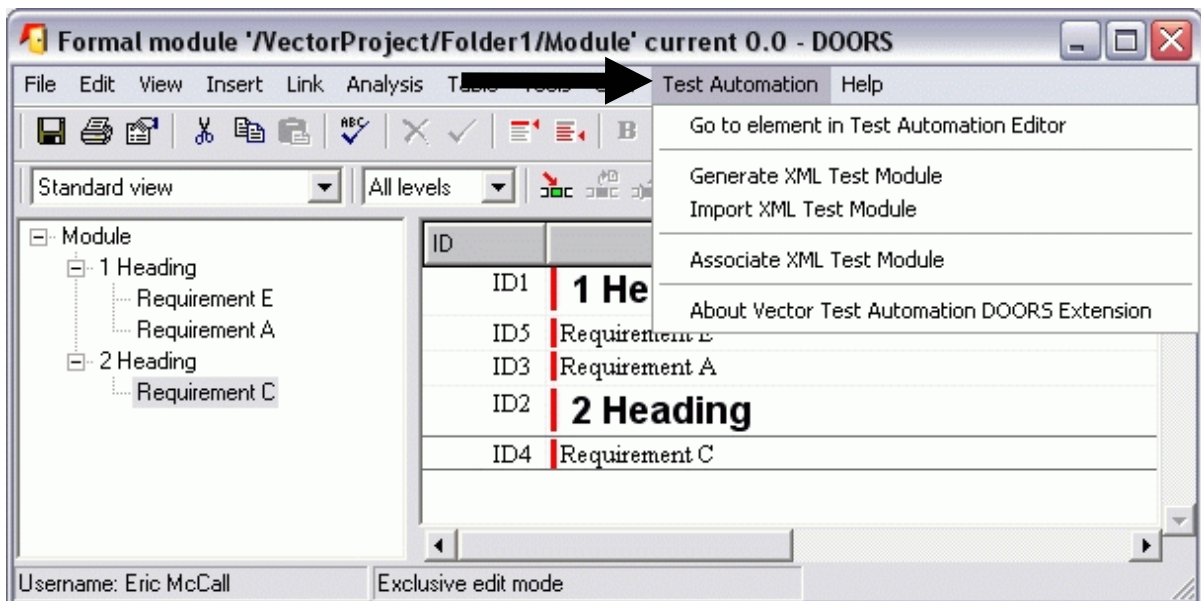


Figure 7: DOORS Module Window with the new Vector Test Automation Menu

## 4.2 Module Generation

### 4.2.1 XML Test Module Creation

This scenario describes in detail how to proceed when an XML Test Module shall be created. The TDM already exists in the DOORS database.

#### Target:

- The XML Test Module is created.
- The TDM 'remembers' the associated XML Test Module, e.g. for synchronization.
- Traceability information is inserted in the XML Test Module.
- Node Objects result in Test Groups.
- Leaf Objects result in Test Case skeletons.
- The object heading is copied to the XML Test Case / Group Title. The DOORS identifier is used when no heading exists.
- The DOORS object text is copied to the XML Test Case / Group Description. Embedded documents are ignored.

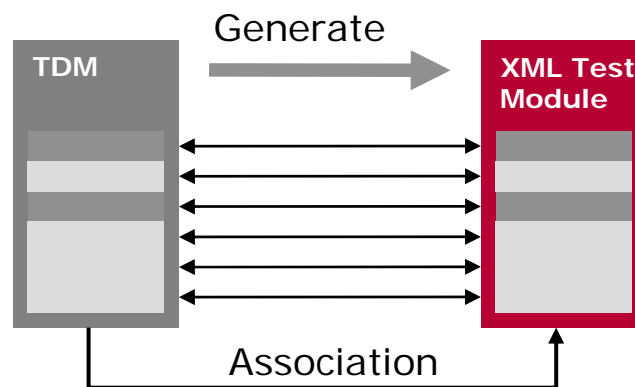
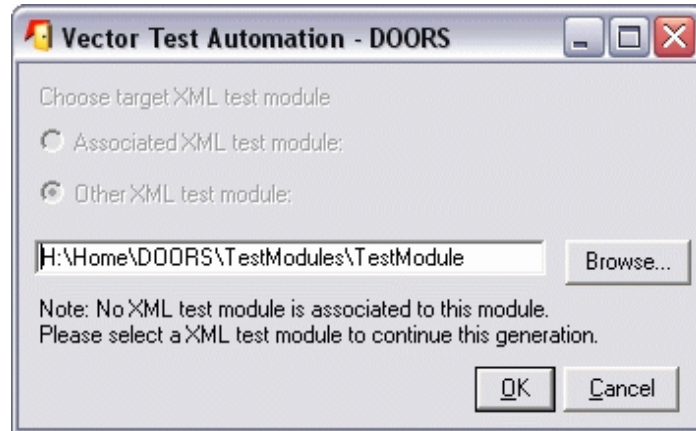


Figure 8: Association and Links between TDM and XML Test Module

#### Steps:

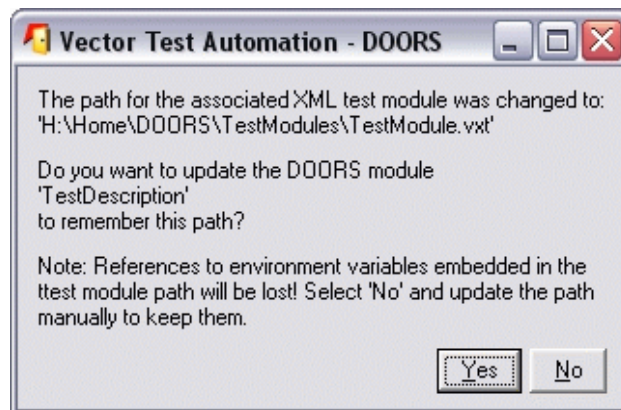
- Open the TDM in the DOORS Module Window and select 'Generate XML Test Module' in the extension menu, see Figure 7.
- Enter the name of the XML Test Module in the following dialog:



**Figure 9: XML File Definition**

Tip: The path definition may contain Windows environment variables in the form %<variable>%

- The next dialog (not shown) configures XML Test Module parameters like 'Encoding'. Here it is sufficient to accept the default settings.
- This is the first time the TDM generates an XML Test Module. It should be associated to the TDM. The association is important when navigating to the XML Test Module or when the XML Test Module needs to be updated.
- The association will be done when you confirm the last dialog with 'Yes':



**Figure 10: Association Confirmation**

Tip: The association can be changed with menu item 'Associate XML Test Module', see Figure 7.

The scenario finishes when the DOORS extension has created the XML Test Module.

## 4.2.2 TDM Creation

In some situations it may be necessary to integrate an existing XML Test Module into the DOORS project database.

### Target:

- The TDM will 'remember' the associated XML Test Module, e.g. for synchronization.
- Test Groups result in Node Objects, Test Cases result in Leaf Objects.
- XML Test Case / Group Titles are copied to DOORS object headings.
- XML Test Case / Group Descriptions are copied to DOORS object text.
- Identifiers and links are synchronized.

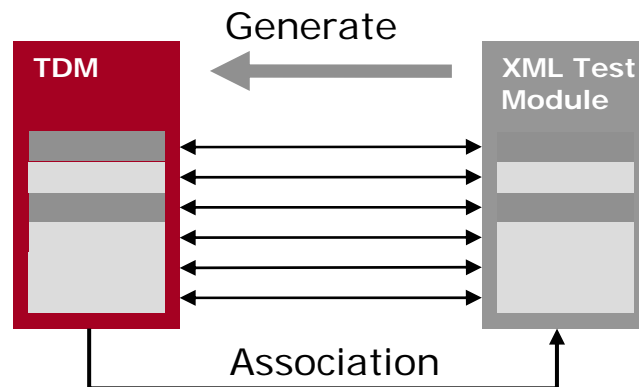
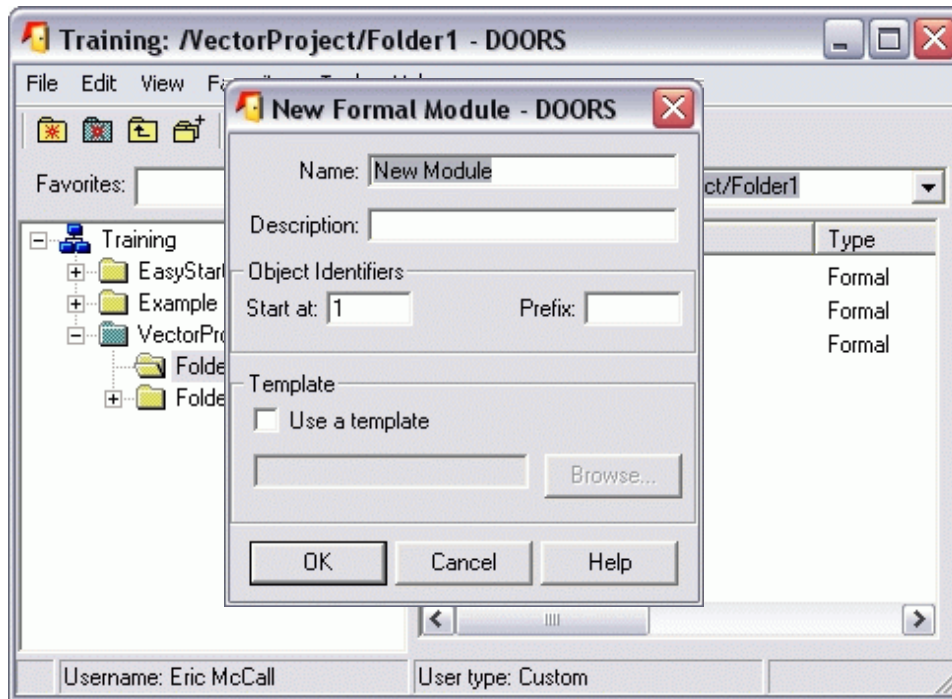


Figure 11: TDM generation from XML Test Module

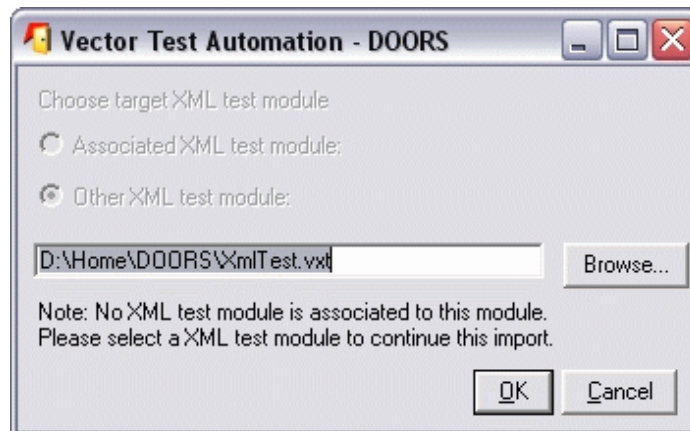
### Steps:

- First create an empty TDM in the DOORS database, see Figure 12. The structure, titles and descriptions from the XML Test Module will be copied to the TDM and you shouldn't enter any information at this point of time.



**Figure 12: Creation of an empty TDM**

- At next open the new TDM and select 'Import XML Test Module'.
- There isn't any associated XML Test Module for the new TDM. Select the XML Test Module in the next dialog:



**Figure 13: The new XML Test Module is associated to the TDM.**

- The content of the TDM will be overwritten with definitions from the XML Test Module. This is OK since the TDM is empty.

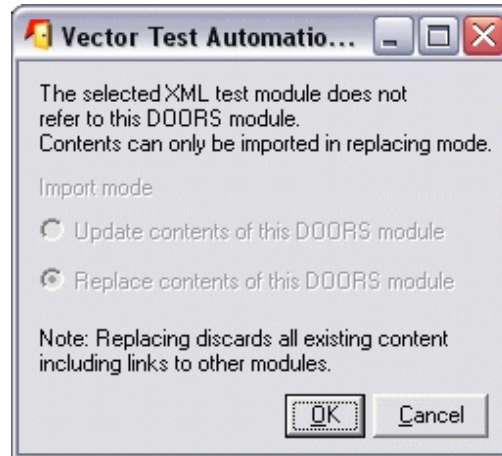


Figure 14: The content of the new TDM is overwritten

- Importing the XML Test Module causes DOORS to create new DOORS objects and identifiers. These identifiers should be written back to the XML Test Module to ensure consistent data in TDM and XML Test Modules:

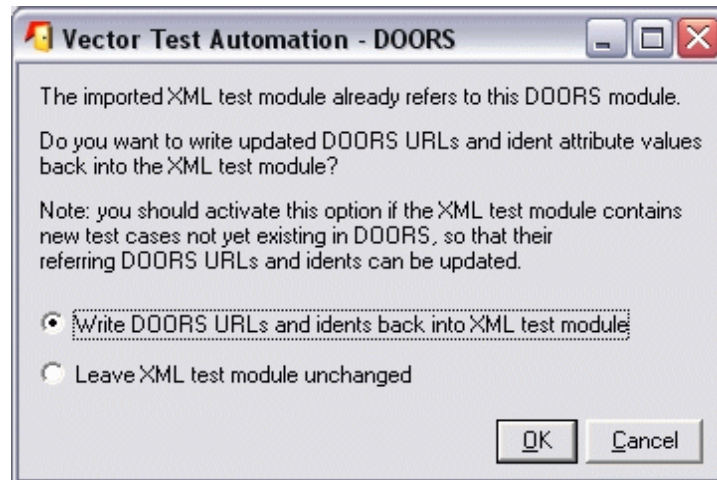


Figure 15: Writing back new identifiers

## 4.3 Navigation

### 4.3.1 Navigation to the associated XML Test Module

#### Target:

- Starting at the TDM the associated XML Test Module shall be opened, e.g. to complete the Test Cases.

**Step:**

- In the DOORS Module Window the menu entry 'Goto Element in Test Automation Editor' is selected, see
- Figure 7. The Test Automation Editor opens the XML Test Module and directly navigates to the selected Test Case respective Test Group.

Tip: CTRL – Clicking on the DOORS link in the Test Automation Editor navigates back to the TDM.

### 4.3.2 Navigation from the Test Report to DOORS

**Step:**

A Test Report as shown in the lower right corner of

- Figure 5 contains links marked with 'DOORS link:'. These links can be used to navigate to TDM and related DOORS Objects. With DOORS internal links you can further navigate to the requirements.

## 4.4 Module Updates

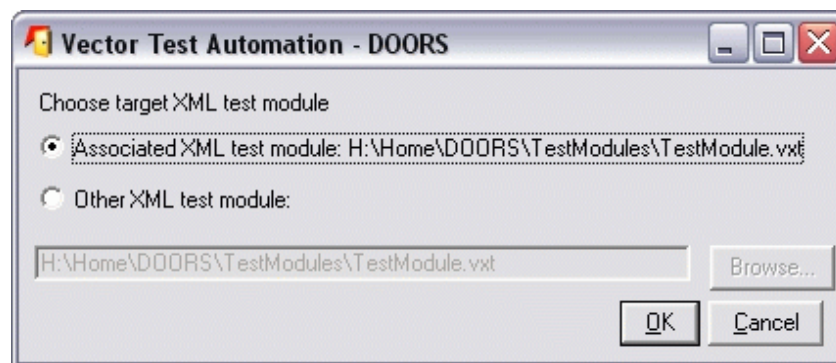
### 4.4.1 XML Test Module Update

**Target:**

- Once a TDM that is already associated to an XML Test Module has been modified, i.e. new tests are added or descriptions are changed, the related XML Test Module needs to be updated. This ensures consistent links and descriptions.

**Steps:**

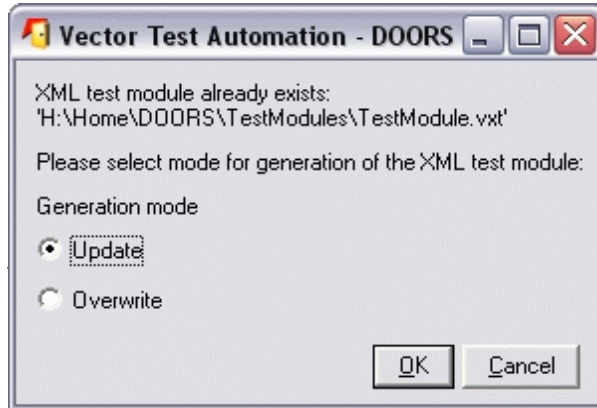
- Open the following dialog with the menu entry 'Generate XML Test Module' in the DOORS Formal Module.
- Select the associated XML Test Module.



**Figure 16: Selecting the target XML Test Module**

- In the last dialog you select if the target XML Test Module shall be updated (recommended) or overwritten. When overwriting the XML Test Module all Test Case implementations will be lost.

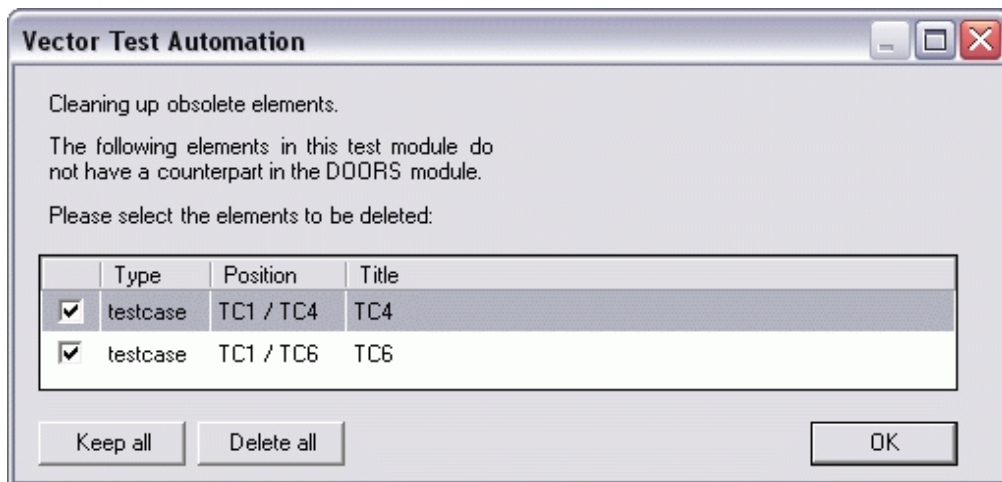
Tip: Before overwriting the XML Test Module a backup (\*.bak) is automatically created.



**Figure 17: Updating or Overwriting the XML Test Module**

The behavior for XML Test Module updates is as follows:

- New DOORS Node and Leaf Objects result in new Test Groups and Test Cases.
- Modified Node and Leaf Objects will be updated in the XML Test Module.
- If its last subordinate Leaf Object was removed, an upper Node Object becomes a Leaf Object and can be converted to a Test Case.
- In case Node and Leaf Objects are deleted in the DOORS Formal Module it can be decided for each item if it shall be removed in the XML Test Module as well, see Figure 18.



**Figure 18: Resolving Conflicts when updating an XML Test Module**

#### 4.4.2 DOORS Module Update

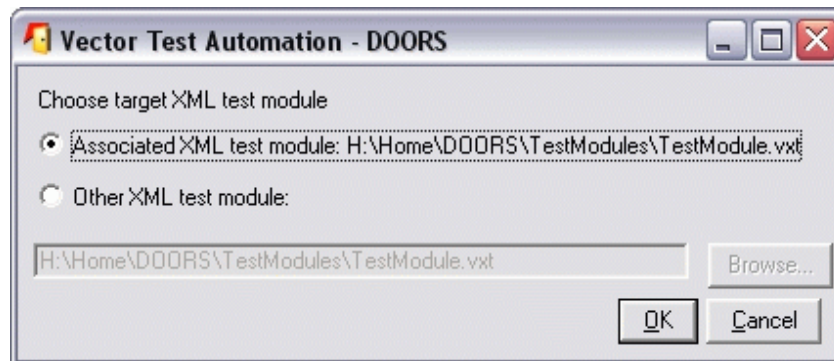
**Target:**

- An XML Test Module that is already associated to a TDM has been modified, i.e. new tests are added or descriptions are changed or corrected. The TDM needs to be synchronized. Synchronisation is only needed when the Test Group / Test Case Titles or the XML Test Module structure has been changed.

Note: Before importing an XML Test Module all current changes in the DOORS Formal Module need to be saved.

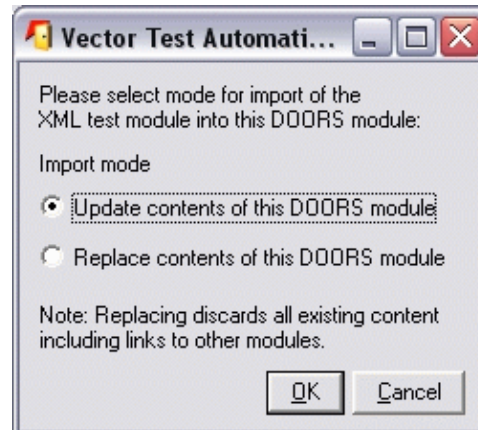
**Steps:**

- Start the update with “Import XML Test Module” in the DOORS Module Window, see Figure 7. The associated XML Test Module shall be used.



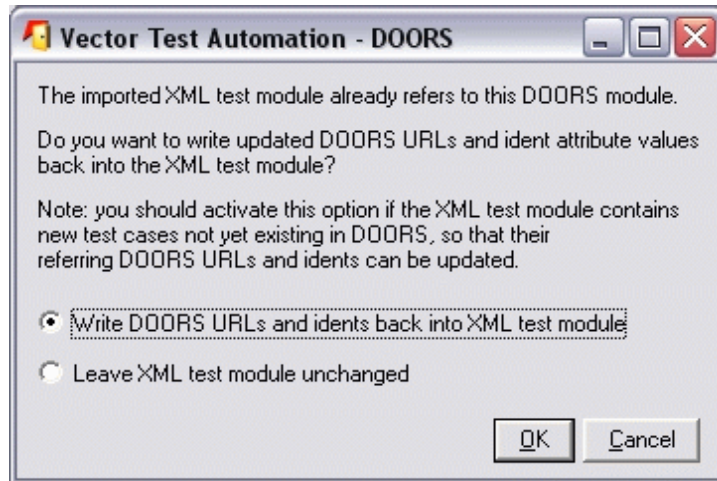
**Figure 19: Selecting the associated XML Test Module**

- In the next dialog you can decide to update the TDM (recommended) or to replace its current content:



**Figure 20: Updating or Overwriting the TDM**

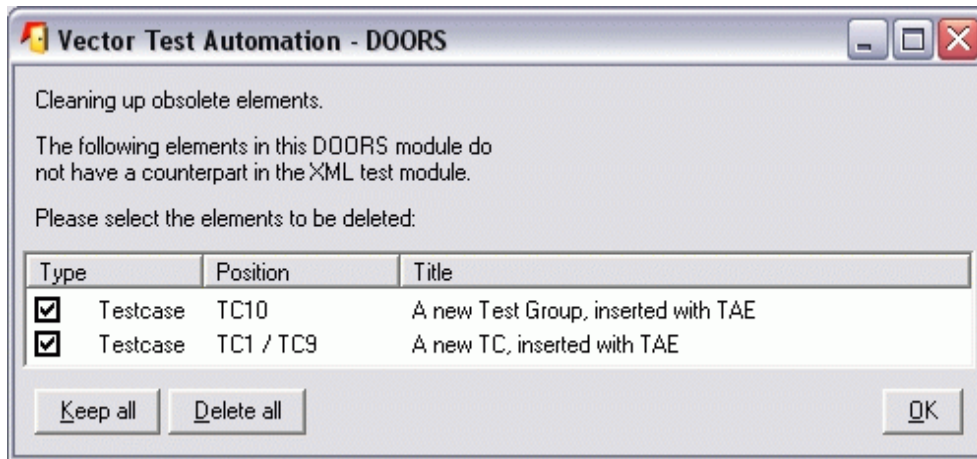
- The following step is very important and keeps the information in the DOORS Module and the XML Test Module consistent. DOORS will create new identifiers when new Test Cases have been defined in the XML Test Module. These new identifiers should be written back to the XML Test Module:



**Figure 21: Writing Identifiers back to the XML Test Module**

The behavior for DOORS Module updates is as follows:

- New Test Groups and Test Cases result in new DOORS Objects. DOORS will generate unique IDs for them.
- The import fails when the XML Test Module contains duplicate IDs.
- In case Test Groups or Test Cases are deleted in the XML Test Module you will be asked if they shall be removed in the DOORS Formal Module as well, see Figure 22.
- Test Case / Test Group descriptions overwrite the corresponding DOORS object text elements, including their embedded documents.



**Figure 22: Resolving Conflicts when updating a DOORS Formal Module**

### 4.5 Adding Traceability to an existing DOORS Database

This scenario describes the migration of an already existing DOORS database and XML Test Modules to a solution that provides traceability.

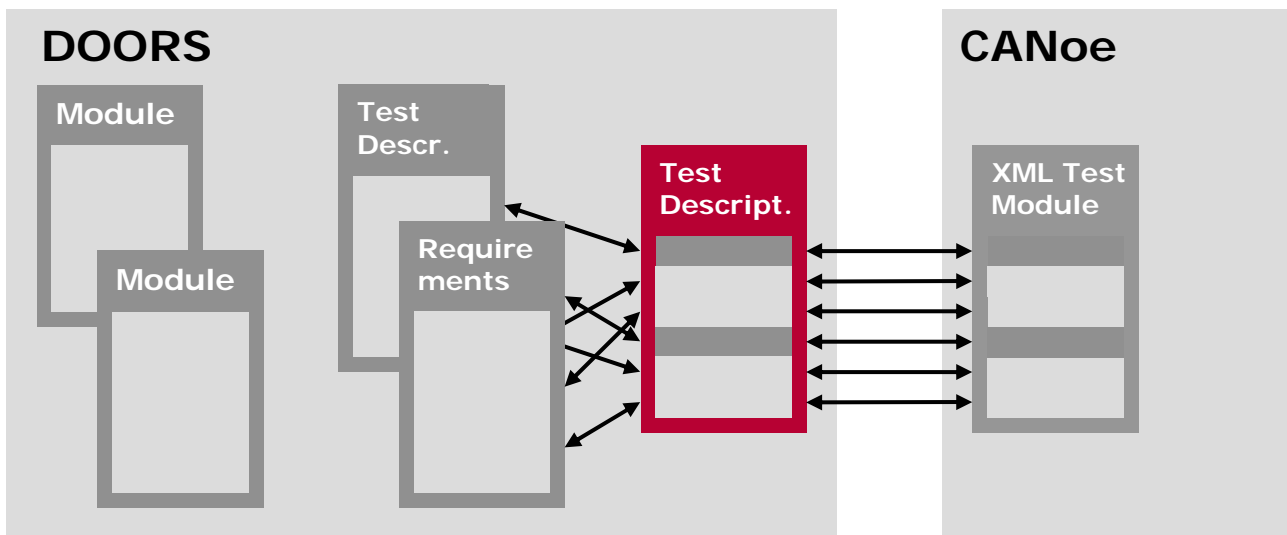
The DOORS database contains arbitrary requirement and / or test description Formal Modules. XML Test modules may exist as well no traceability is provided yet.

**Target:**

- Traceability shall be provided.

**Steps:**

- Generate a new TDM first.
  - either create it manually and afterwards generate the XML Test Module ...
  - ... or generate it from an existing XML Test Module.
- Manually link the new TDM with DOORS internal links to the other Formal Modules.



**Figure 23: Full Traceability added to an existing Project**

## 5 Additional Resources

APPLICATION NOTE:

**AN-IND-1-002** Testing with CANoe

ONLINE DOCUMENTATION:

Test Automation Editor

CANoe

## 6 Glossary

Formal Module	DOORS term for a document that holds information.
Leaf Object	The lowest element in a hierarchy. Leaf Objects will result in Test Cases.
Node Object	Any object that is not a Leaf Object. Node Objects will result in Test Groups.
RQM	Requirements Module. A DOORS formal module that holds requirements.
TDM	Test Description Module. A DOORS formal module that holds test descriptions.
Test Automation Editor	A Vector product to create and maintain XML Test Modules.
Traceability	Capability to retrieve related information
URL	Uniform Resource Locator
XML Test Module	A Vector test description in XML format.
XML	Extended Markup Language

## 7 Contacts

---

### Germany and all countries not named below:

**Vector Informatik GmbH**  
 Ingersheimer Str. 24  
 70499 Stuttgart  
 GERMANY  
 Phone: +49 711-80670-0  
 Fax: +49 711-80670-111  
 Email: info@vector-informatik.de

### France, Belgium, Luxemburg:

**Vector France SAS**  
 168 Boulevard Camélinat  
 92240 Malakoff  
 FRANCE  
 Phone: +33 1 42 31 40 00  
 Fax: +33 1 42 31 40 09  
 Email: information@vector-france.fr

### Sweden, Denmark, Norway, Finland, Iceland:

**VecScan AB**  
 Theres Svenssons Gata 9  
 41755 Göteborg  
 SWEDEN  
 Phone: +46 31 764 76 00  
 Fax: +46 31 764 76 19  
 Email: info@vecscan.com

### United Kingdom:

**Vector GB Ltd.**  
 Rhodium  
 Central Boulevard  
 Blythe Valley Park  
 Solihull, Birmingham  
 West Midlands B90 8AS  
 UNITED KINGDOM  
 Phone: +44 754 9001197  
 Email: info@vector-gb.co.uk

### USA, Canada, Mexico:

**Vector CANtech, Inc.**  
 39500 Orchard Hill Pl., Ste 550  
 Novi, MI 48375  
 USA  
 Phone: +1 248 449 9290  
 Fax: +1 248 449 9704  
 Email: info@vector-cantech.com

### Japan:

**Vector Japan Co. Ltd.**  
 Seafort Square Center Bld. 18F  
 2-3-12, Higashi-shinagawa,  
 Shinagawa-ku  
 Tokyo 140-0002  
 JAPAN  
 Phone: +81 3 5769 7800  
 Fax: +81 3 5769 6975  
 Email: info@vector-japan.co.jp

### Korea:

**Vector Korea IT Inc.**  
 Daerung Post Tower III, 508  
 182-4 Guro-dong, Guro-gu  
 Seoul, 152-790  
 REPUBLIC OF KOREA  
 Phone: +82 2 2028 0600  
 Fax: +82 2 2028 0604  
 Email: info@vector-korea.com

---