IBM Rational DOORS Safety Manual

Revisions for DOORS 9.6 and 9.6.1 releases

v3.0 November, 2014

User	Comment	Date	Version
Karla Ducharme	Started from released DOORS Safety Manual from 2013	November, 2013	DOORS SafetyManual-v2.0.doc
Karla Ducharme	Minor changes to update versions and dates. Added Chapter 11 to include Tool Classification and Error Analysis information for use in company specific tool classification exercises.	November, 2014	DOORS SafetyManual-v3.0.doc

Version History Log

This Program documentation can be used to help Licensee meet compliance obligations, which may be based on laws, regulations, standards or practices. Any directions, suggested usage, or guidance provided by this Program documentation does not constitute legal, accounting, or other professional advice, and Licensee is cautioned to obtain its own legal or other expert counsel. Licensee is solely responsible for ensuring that Licensee and Licensee's activities, applications and systems comply with all applicable laws, regulations, standards and practices. Use of this Program documentation does not guarantee compliance with any law, regulation, standard or practice

License Agreement

The information in this publication is subject to change without notice, and IBM assumes no responsibility for any errors which may appear herein. No warranties, either expressed or implied, are made regarding IBM DOORS software including documentation and its fitness for any particular purpose.

Trademarks

Table of Contents

1 Purpose

This manual provides support for software tool classifications of IBM Rational DOORS (referred to as "DOORS" in this document) according to **[ISO 26262]** and **[IEC 61508]**. This manual describes the features provided by DOORS, and errors which may occur in using these features. By means of some sample use cases it demonstrates how to determine a DOORS Tool Confidence Level (TCL) suitable for specific development processes of safety relevant products.

This manual is does not serve as an introduction to DOORS. The reader is supposed to have some basic knowledge of DOORS in order to make best use of this manual.

- Section 2 defines the scope of the certification, indicating what features of DOORS are covered by the certificate and what features are not covered. This section also lists the responsibilities of the user in using the information of this document.
- Section 3 compiles the relevant information about the tool DOORS. In addition, it proposes a method by which the information of this document can be used to perform the analysis according to the safety standards.
- Section 4 provides guidance on how to use DOORS safely.
- Section 5 specifies the features of DOORS that are in the scope of the certificate.
- Section 6 gives additional information on recommended usage of DOORS.
- Section 7 mentions some sample use cases. This is not an exhaustive list, it is the duty of the DOORS user to define use cases according to their process.
- Section 8 summarizes how the DOORS features considered relate to the safety standards ISO 26262 and IEC 61508.
- The remaining sections contain the mappings to the relevant standards, the list of references, the list of figures, and generic Tool Classification and Error Analysis information.

2 Scope of the certification

In this section the scope of the certification is defined. Not all features of DOORS are included in the certificate. We briefly list what features of DOORS are covered. We also state what features of DOORS are not covered.

This section concludes in describing how users can make use of the information compiled in this document in order to evaluate their way of using DOORS according to the safety standards.

2.1 Certified features of DOORS

The certificate covers the versions 9.6 and 9.6.1 of the DOORS server and the DOORS client, and DOORS Web Access. DOORS Next Generation functionality is not covered by the certificate.

The following features of DOORS are in scope of the certificate. More details will be provided in section 5.

- Features at database level
  - Database Explorer
  - Finding data in a DOORS database
  - Projects and folders
- Features for documents
  - Formal modules
  - Editing formal modules
  - Objects
  - Attributes
  - OLE objects and pictures
  - Tracking changes to DOORS data
  - Discussions
- Features for displaying information
  - Controlling the data that is displayed in formal modules
- Features for links
  - Standard links, external links, and collaboration links
  - Link modules, linksets, and linkset pairings
  - Links and traceability
  - DOORS URLs
- Features for data exchange
  - ReqIF
  - Data partitions
  - Exporting data from formal modules
  - Exporting data from formal modules using Rational Reporting for Document Generation
  - Printing and reports
  - Importing data into formal modules
- Additional features
  - Authenticate
  - Access rights
  - Baselines
  - Keyboard shortcuts
- Features for customizing DOORS
  - Using DXL (the DOORS Extension Language)

- Non-critical Features
    - Messages
    - Checking database integrity

Note that the feature "Using DXL" covers the core DXL as provided by DOORS. Programs written using this DXL are outside the scope of this safety manual.

DOORS has a client-server architecture.

- A single process, the DOORS database server, accesses the data directory in the file system and executes requests from DOORS clients.
- The rich client processes the data received from the database server locally and returns the data when the user saves.

DOORS Web Access (DWA) adds two components to this architecture: a web server, called DWA server, and a message broker, called DWA broker.

- The DWA server processes the data for display in a web browser.
- The DWA broker controls the communication, i.e. the broker routes answers from the database to the browser requesting the data.
- Data received from the database are processed by the DWA interoperation server, which is a DOORS client running in a server mode, i.e. without user interface.
- The DOORS database server retrieves data from the data repository as requested by the DWA interoperation server.

Owing to the fact that the DWA interoperation server and the rich DOORS client are identical, and both are communicating with the DOORS database server, errors that apply to client-server systems, like e.g. loss of connection or no connection, were considered in IBM’s error analysis of DOORS.

Note that the features supported by DWA constitute a subset of the list for the rich client. Some features, like e.g. "Link modules, linksets, and linkset pairings" which subsumes the functionality needed for administration of links, are not supported by DWA, so they must not be associated to use cases describing DWA.

2.2 Other features of DOORS

DOORS provides some features in addition to those listed in the preceding section which are not analyzed in this manual. We have excluded features that are not recommended for use in safety critical context in new implementations. As there is no reason in principle to exclude these features, users might complete an individual evaluation of those features if they desire to use them.

The following DOORS features are not included in the scope of the tool evaluation.

- Descriptive Modules
- Change Proposal System
- Test Tracking Toolkit

User management is considered out of scope. DOORS user accounts may be managed by standard DOORS functionality, or by employing the Rational Directory Server (RDS), an application based on LDAP v3 for managing user information for all Rational lifecycle management tools in one place. Errors in user management may result in log in failure, but will not directly result in data errors as users are always subject to the access rights set up, a feature which is analyzed in this manual.

Integrations between DOORS and other Rational or third party tools are out of scope. Integrations require an extended analysis, covering the tool integrating with DOORS and the integration application as well.

User programs in DXL (DOORS extension language) are out of scope. The base DXL commands and the API are covered, but user programs need to be analyzed. In case a program has an impact, its TCL must be determined in terms of use cases, features, errors and checks to prevent or detect errors with a LOW or MEDIUM error detection probability

2.3 Responsibilities of the user

This manual compiles the information needed to evaluate the use of DOORS in safety critical context. The safety standards require the users to perform the evaluation for each specific employment, so there can be no general statement that any tool is safe to use under all circumstances.

The process of evaluating a tool is summarized in the following diagram.

This manual is no replacement for the **[DOORS User Manual]**, nor does it replace adequate education of the staff. The Safety Manual addresses experienced users as not all DOORS terms will be detailed. It is the responsibility of the customer to provide suitable enablement to the staff using DOORS, as both tool knowledge and knowledge about the development process is concerned.

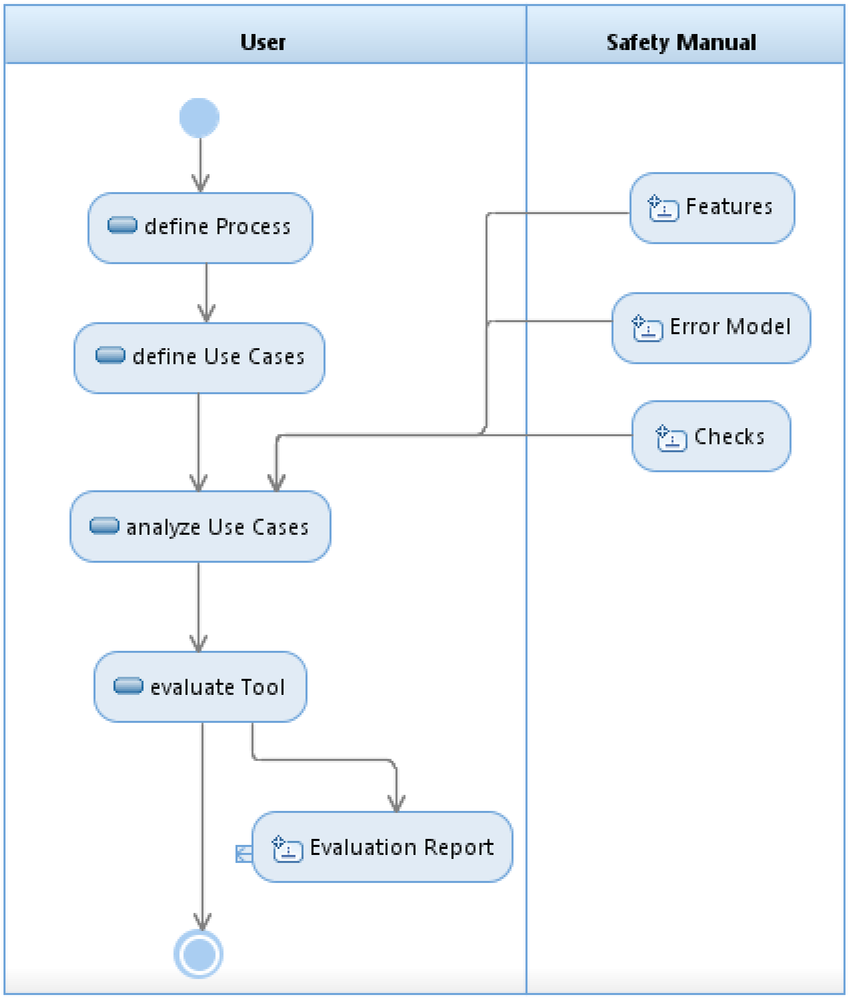


Figure 1: Preparing a tool evaluation report

First, the user has to define a process describing the use of DOORS and derive the use cases relevant for safety critical context. The user then analyzes what features provided by the tool are employed by each use case. The DOORS features are described in this manual. Finally each use case must be classified according to the classification schema defined by the safety standard.

According to ISO 26262 (8-11.4.10) it is the duty of the user to perform a confirmation review re-validating any TCL that is given in this manual. For each project this review verifies

- the correct evaluation of the level of confidence of the software tool, and
- the appropriate qualification of the software tool in accordance with its level of confidence.

In a specific situation the assessment may be different from the one given here because assumptions that were made do not hold in that situation. So the user has to check the assumptions, and the user must make sure that the assumptions which are adopted are reflected in the process.

3 Overview

This section assembles some background information on DOORS. It states what Version of DOORS the analysis applies to, and it gives reference on where to find manuals as well as on the support process. In this section we also provide guidance on how to use DOORS safely. The section concludes by sketching the method that was used to evaluate DOORS.

3.1 Purpose of using DOORS

DOORS is a requirements management tool that makes it easy to capture, trace, analyze, and manage changes to information. Control of requirements is key to reducing costs, increasing efficiency, and improving the quality of products or services. With its own built-in database, DOORS provides a rich set of features to help you capture and manage requirements.

DOORS supports everyone in an organization that participates in and contributes to the requirements management process by

- providing access to the requirements database using a rich client or a web browser
- supporting the capture, analysis, derivation, and review of requirements
- supporting links from requirements to design items, test plans, test cases, and other requirements for easy and powerful traceability
- facilitating direct collaboration of business users, marketing, suppliers, systems engineers, and business analysts through requirements discussions
- directly involving suppliers and development partners in the development process employing the Requirements Interchange Format (ReqIF).
- integrating with engineering design and lifecycle tools using the Open Services for Lifecycle Collaboration (OSLC) specifications for requirements management, change management and quality management. This includes Rational Requirements Composer, Rational Rhapsody, Rational Team Concert, Rational Design Manager, Rational Quality Manager, Rational Focal Point, and Rational System Architect, and also many third-party tools, providing a comprehensive traceability solution.

Note: RegIF was formerly known as RIF. The Object Management Group **[OMG]** took over the management and evolution of the Requirements Interchange Format from **[HIS]**. The Acronym RIF conflicted with the W3C`s Rule Interchange Format specification so the acronym has been changed to ReqIF.. ReqIF 1.0 is the direct successor of the ProSTEP iViP recommendation RIF1.2

The reference process analyzed in section 6 uses the tool in several process phases for similar purposes, all related to requirements management, ranging from release planning to system test.

3.2 Version of DOORS

Both DOORS and DOORS Web Access display their version on the splash screen before logging in. You can check what version of both DOORS client and DOORS server you have installed using "About DOORS..." in the help menu.

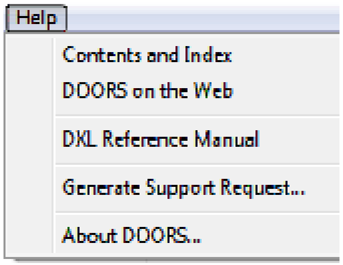
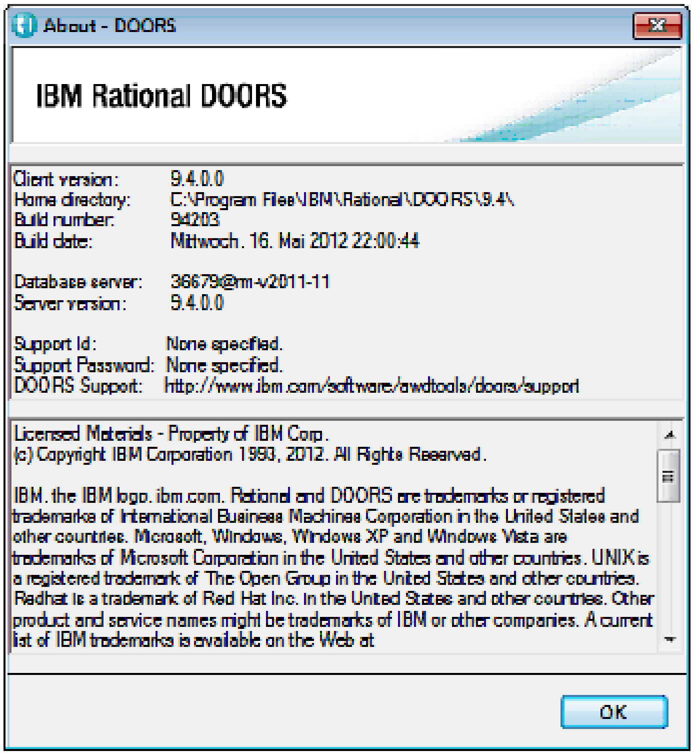


Figure 2: The DOORS help menu



Client version:	9.4.0.0
Home directory:	C:\Program Files\IBM\Rational\DOORS\9.4\
Build number:	94203
Build date:	Mittwoch. 16. Mai 2012 22:00:44
Database server:	36679@m-v2011-11
Server version:	9.4.0.0
Support Id:	None specified.
Support Password:	None specified.
DOORS Support:	http://www.ibm.com/software/awdtools/doors/support

Licensed Materials - Property of IBM Corp.
(c) Copyright IBM Corporation 1993, 2012. All Rights Reserved.

IBM. the IBM logo. ibm.com. Rational and DOORS are trademarks or registered trademarks of International Business Machines Corporation in the United States and other countries. Microsoft, Windows, Windows XP and Windows Vista are trademarks of Microsoft Corporation in the United States and other countries. UNIX is a registered trademark of The Open Group in the United States and other countries. Radhat is a trademark of Red Hat Inc. in the United States and other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at

Figure 3: The "About DOORS" dialog

In DOORS Web Access you can check the version in the "About DOORS Web Access" item of the help menu. You access the help menu using the question mark in the top right of the page.

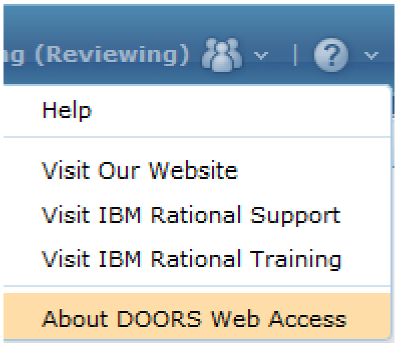
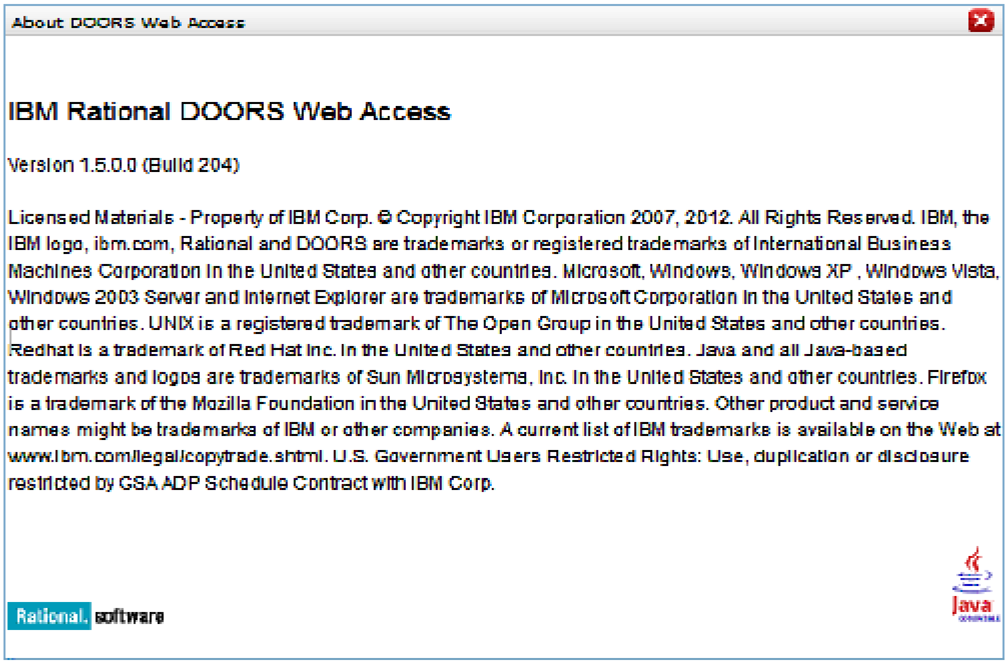


Figure 4: The DOORS Web Access help menu



About DOORS Web Access

IBM Rational DOORS Web Access

Version 1.5.0.0 (Build 204)

Licensed Materials - Property of IBM Corp. © Copyright IBM Corporation 2007, 2012. All Rights Reserved. IBM, the IBM logo, ibm.com, Rational and DOORS are trademarks or registered trademarks of International Business Machines Corporation in the United States and other countries. Microsoft, Windows, Windows XP , Windows Vista, Windows 2003 Server and Internet Explorer are trademarks of Microsoft Corporation in the United States and other countries. UNIX is a registered trademark of The Open Group in the United States and other countries. Redhat is a trademark of Red Hat Inc. in the United States and other countries. Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. In the United States and other countries. Firefox is a trademark of the Mozilla Foundation in the United States and other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml. U.S. Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Figure 5: The "About DOORS Web Access" dialog

3.3 Reference manual

For this document, the terms “Reference Manual” and “User Guide” refer to the same artifact. The DOORS reference manuals may be found online at the **[IBM Product Information Center]**. The DOORS reference manual for your specific version of DOORS, referenced on this page, can be found at the Rational **[DOORS Information Center]**.

The DOORS reference manual can also be accessed from the DOORS client, using the "Contents and Index" item from the DOORS help menu.

The start page of the reference manual provides links to additional important documentation, like RSS feeds, and resources at http://www.ibm.com, and featured documents.

- the "DXL reference manual" documents the scripting language DOORS supports
- "Get it right the first time" gives advice on how to write requirements efficiently
- the "DOORS API manual" describes how to integrate DOORS with other applications

The section "Overview and installing" of that page points to several collections of useful information.

The release notes can be accessed by the link "Rational DOORS readme".

There is a general overview of DOORS.

A short introduction on requirements mentions the document "Get it right the first time".

Some information on installing DOORS is listed at "Preparing to install".

There is an option to print the contents of the DOORS help, or to create PDF files with contents of the DOORS help.

The menu "DOORS on the Web" refers to the web site **[DOORS Communities]**. This web site offers access to 5 forums addressing various topics.

- IBM Rational DOORS DXL forum on developerWorks
- IBM Rational DOORS administration forum on develeoperWorks
- IBM Rational DOORS general discussion forum on developerWorks
- IBM Rational DOORS integrations forum on developerWorks
- Rational RFE Community (Submit feature requests)

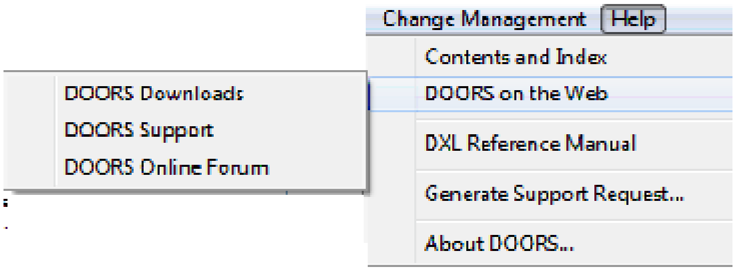
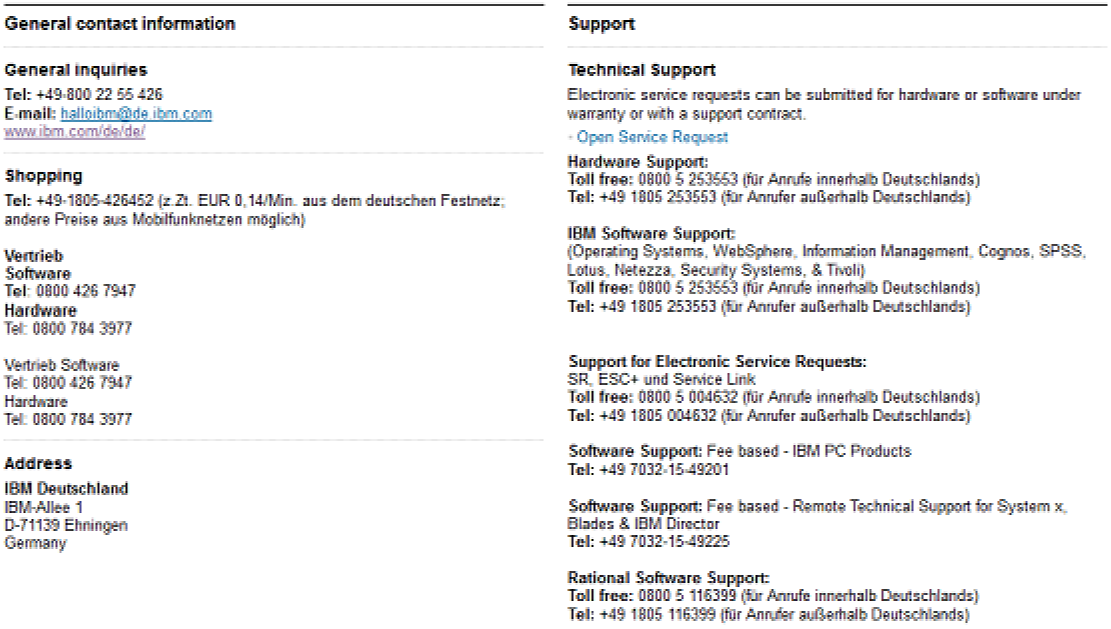


Figure 6: The menu "DOORS on the Web"

3.4 Support information

IBM provides many resources to help our customers. Our main support page at **[IBM Support Home]** has lists of support resources, contact information, and other related content. If you expand the "Contact Support" field on the right hand side of this page there is a link for a "Directory of worldwide" contacts. This will take you to a list of countries. Select your country for the contact information that is specific to you. An example of how this information looks is in the figure below.



General contact information

General inquiries
Tel: +49-800 22 55 426
E-mail: halloibm@de.ibm.com
www.ibm.com/de/de/

Shopping
Tel: +49-1805-426452 (z.Zt. EUR 0,14/Min. aus dem deutschen Festnetz; andere Preise aus Mobilfunknetzen möglich)

Vertrieb
Software
Tel: 0800 426 7947
Hardware
Tel: 0800 784 3977

Vertrieb Software
Tel: 0800 426 7947
Hardware
Tel: 0800 784 3977

Address
IBM Deutschland
IBM-Allee 1
D-71139 Ehningen
Germany

Support

Technical Support
Electronic service requests can be submitted for hardware or software under warranty or with a support contract.
· Open Service Request

Hardware Support:
Toll free: 0800 5 253553 (für Anrufe innerhalb Deutschlands)
Tel: +49 1805 253553 (für Anrufer außerhalb Deutschlands)

IBM Software Support:
(Operating Systems, WebSphere, Information Management, Cognos, SPSS, Lotus, Netezza, Security Systems, & Tivoli)
Toll free: 0800 5 253553 (für Anrufe innerhalb Deutschlands)
Tel: +49 1805 253553 (für Anrufer außerhalb Deutschlands)

Support for Electronic Service Requests:
SR, ESC+ und Service Link
Toll free: 0800 5 004632 (für Anrufe innerhalb Deutschlands)
Tel: +49 1805 004632 (für Anrufer außerhalb Deutschlands)

Software Support: Fee based - IBM PC Products
Tel: +49 7032-15-49201

Software Support: Fee based - Remote Technical Support for System x, Blades & IBM Director
Tel: +49 7032-15-49225

Rational Software Support:
Toll free: 0800 5 116399 (für Anrufe innerhalb Deutschlands)
Tel: +49 1805 116399 (für Anrufer außerhalb Deutschlands)

Figure 7: Sample contact information for Germany

Note, you can contact IBM via phone or web based methods.

The *IBM Software Support Handbook* is a document that describes the IBM support processes and how different kinds of support questions are handled. It is available on the web at **[IBM Support Handbook]**.

IBM Rational also runs application specific support sites. The DOORS support site is located at **[DOORS Support]**. Reference to this web site is given in the DOORS menu of both the database explorer and the module explorer at "> Help > About DOORS", cf. above.

The help menu also provides the option to directly generate a support request, and it provides access to the DOORS support site in the sub-menu "> DOORS on the web > DOORS Support", cf. above.

The following sections (3.4.1, 3.4.2, and 3.4.3) summarize parts of the *IBM Software Support Handbook* . That information is provided here for convenience. Please refer to the source document for more complete descriptions of these and other support topics.

3.4.1 How a service request is handled

Requests that are raised will be logged into the IBM problem management system and assigned a unique Service Request (SR) or Incident/Support Case ID. This number must be used in any future communication on the issue with the support center. The SR, Incident, or Support Case is routed to a resolution team for handling.

Any initial communication with the support organization is through a Level-1 representative, who searches for problems with similar symptoms. If the problem is found to be already known, and a fix has been devised for it, availability of corrective service software will be advised. If the search is unsuccessful, the problem is passed on to a Level-2 representative to start a more detailed investigation of the cause.

The Level-2 representative may ask for more details or documentation. Based on the information supplied, the investigation then carried out can determine whether the cause of the problem is new to the support organization, or is already known.

If the Level-2 need additional assistance Level-3 support will be involved. Level-3 is development. They are responsible for providing Level-2 with information or guidance to resolve a customer issue in an escalation. Level-3 is also responsible for creating fixes for defects.

3.4.2 Preventing Problems: Known problems and fix notification

Every time that a defect is found in the code either by external customers, or internally by IBM staff which will impact our customers, an **Authorized Problem Analysis Report** (called APAR document) must be created so that customers can be alerted to the problem and can determine which Fix Pack will ultimately include the fix for this bug. It is very important to understand that APARs are external documents.

IBM recommends the installation of Preventive Service Packages to proactively avoid impacting problems caused by software defects already known and corrected by IBM.

Preventive Service Packages are updated frequently. In a stable environment where problems are never encountered, not every package needs to be installed. However, periodic installation is recommended, since fixes are nearly always built at the latest maintenance level. Keeping fairly current with service will reduce the volume of change required in case a fix should be needed. The particular time when to update must be determined by the user, however, pondering the benefits of an update against the risk of disturbing projects in critical phases, for instance.

Fix Central makes it simpler to find the fixes from IBM, you can search by product, operating system, release, or even by APAR ID or fix ID. IBM product teams are in the process of adopting this delivery option, so if the fix you want is not available through Fix Central, it will be available on the product support page. Fix Central can be found at **[IBM Fix Central]**. Please note that not all fixes are made available at both Fix Central and the product support page. In order to be sure you get the most recent fix you have to visit both sites.

A list of open APARs is available at **[IBM Rational Open APARs]**. The list of APARs or defect IDs that are fixed in a certain DOORS release can be found on the web, too. The corresponding web site is included in the release notes of that version, which can be downloaded as part of the installation package. The release notes are also referred to in the section "Overview and installing" on the DOORS help site. For the current version of DOORS the list of fixes can be accessed at **[DOORS Fix List]**.

3.4.3 Stay Informed

As it is hard to stay informed, IBM have a system that will email customers when new fixes or alerts are released. Customers create a profile indicating what kinds of information they are interested in, so they only receive the information they are looking for.

Support Subscriptions can be set up via *My Notifications* at **[IBM Support Subscriptions]**.

My Notifications allows customers to receive security advisories and alerts that maintenance fixes are available. The website presents the complete list of what can be signed up for. The basic free service is extremely helpful, but the capabilities of this service can be expanded.

3.5 Evaluation Method and ISO 26262 Requirements

This section describes both the requirements and a method to determine the level of confidence for the tool chain as used in the development process. The ISO 26262-8, 11 defines the requirements for the TCL that is used to determinate the required tool qualification techniques. In this section the method is introduced abstractly, sections 5 and 6 apply the method to DOORS.

The ISO 26262-8 recommends various steps and documents in order to establish “Confidence in the use of software tools” (see Figure below). These steps are explained shortly in the following subsections.

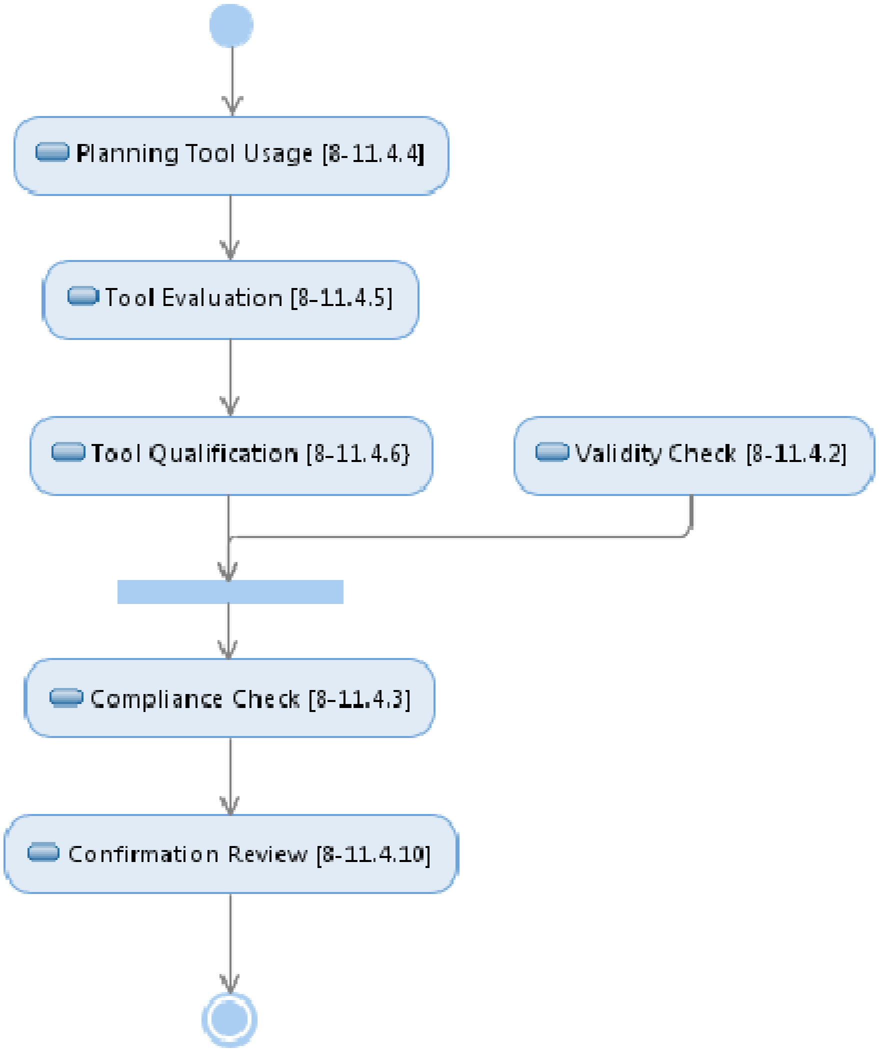

Planning Tool Usage [8-11.4.4]
Tool Evaluation [8-11.4.5]
Tool Qualification [8-11.4.6}
Validity Check [8-11.4.2]
Compliance Check [8-11.4.3]
Confirmation Review [8-11.4.10]

Figure 8: ISO 26262-8 "Confidence in use of software tools"

This manual is intended to be used to help to create a "Tool Evaluation Report" and therefore only covers parts of the process. However, the other parts of the process are described briefly also to indicate further qualification activities.

3.5.1 Planning Tool Usage

All tools to be used for the development of a safety related product need to be planned first (ISO 26262-8 11.4.4, ISO 26262-6 5.5.4). Tool planning involves the setting of tool application guides, which are needed to capture or refer to information relevant for the evaluation or use of a software tool. The definition and description of the use cases for a software tool are most important for the tool evaluation.

3.5.2 Tool Evaluation

The tool evaluation analyzes the usage of the tool and determines the tool confidence level (TCL) for the tool. The result of the tool evaluation is a Tool Criteria Evaluation Report, which according must contain the following information (ISO 26262-8 11.4.5):

- the intended purpose
- the inputs and expected outputs
- the environmental and functional constraints
- an analysis of possible malfunctions
- an analysis of the Tool Confidence Level (TCL)

To determine the tool confidence level for a tool the ISO 26262-8 requires two evaluations:

1) Determine the tool impact (TI)
2) Identify the potential tool errors and estimate a tool error detection (TD) for measures that either detect or prevent these errors.

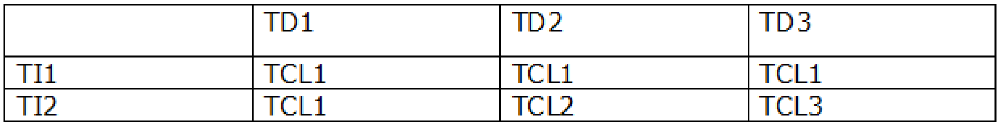
A tool has no impact (TI1) if there is no possibility that a potential tool error can introduce errors or fail to detect errors in a safety related part of the product. If no arguments can be found against each of the conditions it must be assumed that a tool has impact (TI2), which means that potential tool errors may affect the safety of the developed product.

For all tools with potential impact (TI2) all relevant use cases have to be analyzed as follows: All potential errors have to be considered and if possible measures that can detect or prevent these errors need to be assigned. For each measure a qualitative tool error detection (TD) probability has to be assigned:

- TD=1 if the probability to detect or prevent the error is HIGH,
- TD=2 if the probability to detect or prevent the error is MEDIUM and
- TD=3 in all other cases (LOW or unknown probability).

If several detection or prevention options are available for one error, the one with the best detection or prevention probability may be used. In case there are multiple potential errors for one tool or use case, the one with the worst detection or prevention probability determines the TD for the tool or use case.

Once the TI and TD levels have been estimated the required tool confidence level is determined by ISO 26262 according to the following table (see figure 9).



	TD1	TD2	TD3
TI1	TCL1	TCL1	TCL1
TI2	TCL1	TCL2	TCL3

Figure 9: Tool Confidence Levels according to ISO 26262

For each DOORS feature IBM used the HAZOP technique. Every artifact written was confronted with the following keywords “Too much”, “Too less”, “Wrong”, “Too early” and “Too late”. For example, in the case of applying a filter on data, if we confront the output artifact “filtered data set” with the key word “Too Little” we can come up with the potential error “Not all data shown”.

DOORS features where user or other errors would be detected with high confidence in a normal safety process were given a generic Tool Confidence Level (TCL) of 1. DOORS features where an error would not be detected with high confidence were given a generic TCL of 3.

The DOORS error model and the validation suites for any feature with a potential error with error detection of Low or Medium were reviewed in an independent assessment by the TÜV SÜD. The validation test suites were reviewed as their completeness and robustness per their associated tool operational requirements. These tests were executed for the referenced DOORS releases. Consequently, all features defined in the scope of this document and the accompanying TÜV SÜD "Fit for Purpose" certificate can be used on projects up to ASIL "D" or SIL "3".

A summary of the generic (out of the context of a specific use case) error model is provided in Section 11 as a reference.

To obtain more information on the error analysis done for DOORS, please contact your local IBM Rational support.

3.5.3 Tool Qualification

Depending on the automotive safety integrity level (ASIL) of the item being developed and the tool confidence level the ISO 26262 recommends whether and how a tool needs to be qualified. For ASIL D and tools with TCL2 or TCL3 the ISO 26262 recommends "Tool Validation" or "Development of the tool according to a safety standard" as qualification methods. The goal of tool qualification is to show the absence of the potential errors identified during the tool evaluation. The result of tool qualification is a Tool Qualification Report, which must contain the information as required by ISO 26262 8-11.4.6.2 and 8-11.5.2.

3.5.4 Validity Check

Tool evaluation results, that is the TCLs obtained from the tool qualification process and the qualification reports may be reused in various development projects. However, if this is done, the validity of this predetermined information for the current project needs to be checked.

3.5.5 Compliance Check and Confirmation Review

Each software tool used within a safety related development project needs to be checked in a "tool compliance check" whether the tool is used as specified in the Tool Evaluation Criteria Report.

The confirmation review checks the correctness of the Tool Evaluation Criteria Report, and it checks that the tool qualifications are satisfied by the way this specific project uses the tools. If, for example, the tool evaluation relies on certain activities or checks being implemented in the development process, then it must be verified that these activities or checks are actually implemented in the current process.

4 Practices to use DOORS safely

This section addresses several topics that might impair a safe operation of DOORS, including the installation of the tool and a backup strategy. The configuration of the tool as well as user management is discussed, and we emphasize the usefulness of a proper information architecture.

4.1 Installation

There are several types of installation for DOORS. This section summarizes the typical installation on Windows. For different scenarios please refer to the corresponding section of the online documentation at the **[DOORS Information Center]**.

4.1.1 Pre-installation tasks

Make sure that you meet the system requirements before you start installing DOORS.

- Check that you meet the **[DOORS System Requirements]**, else refer to the release notes.
- Make sure licenses are available. If you have procured floating licenses, a Rational License Key Server must be installed on your network. For the client installation you need the name of the machine hosting this service and its port number. Otherwise the license files for all client machines must be at hand.
- Make sure to log in as the local Administrator on the machine you are installing DOORS, and not a user with Administrator privileges.
- Shut down all other applications. In particular, shut down Microsoft Office applications, including the Microsoft Office toolbar.

Make sure that you download the correct version of the installers. The system requirements indicate what versions of the components are compatible.

Consider doing the installation in a test environment before updating the production environment in order to make sure that the new version fully supports your development process.

4.1.2 Installation tasks

There are two installation tasks: install one or several DOORS database servers, and install one or several DOORS clients. Both are Windows standard installations which guide the user, and for both the installation path must be entered.

The installer of the **DOORS database server** in addition requires the port number for the service (defaulting to 36677) and the directory containing the data (e.g. C:\DOORSdata).

- Make sure that the service which is created (named DOORS DB Server 9.x to correspond to the version of DOORS being deployed) is started. The service will fail to start if the port is already in use or if the directory specified does not exist.
- Multiple database services can be set up on the same machine, so it is not necessary to physically install separate server processes on one machine. Refer to the section "Using the command line to install and remove Rational

DOORS database server services" of the **[DOORS User Manual]** for details of how to do this.

- If you install more than one server process on a single machine, each server process must have a different data directory and port number.

The installer of the **DOORS client** in addition needs the port numbers and the full qualified names of the machines hosting the DOORS database server and the Rational License Key Server, if you use floating licenses. Otherwise you must provide the path to the license file.

- Hint: The installer expects information on the license server to be entered in the form *port@servername*, e.g. *27000@myServer.ibm.com*.
- A single installation of the DOORS client can be directed to separate databases by adding command-line parameters to Windows shortcuts, so it is not necessary to install multiple DOORS clients on one machine. Refer to the section "Starting Rational DOORS from a shortcut" of the **[DOORS User Manual]** for details of how to do this.

For both installation tasks check the version information that is displayed on the splash screen as well as on several dialogs.

Please note that in upgrading from a former DOORS version two scenarios must be distinguished, cf. section "Upgrading from a previous version" of the **[DOORS User Manual]**.

- Installing a later minor release (upgrading from 9.2 to 9.5) of the DOORS server or the DOORS client will remove the earlier version.
- Installing a later major release (upgrading from 8.1 to 9.5) of the DOORS server or the DOORS client will retain the earlier version. In this scenario your must first log in as Administrator to the new database server. Generally, you can only access a database if server and client share the major version.

In order to prevent unintentional outages of the DOORS system make sure to plan any update appropriately. The update plan must consider the migration of both database server and client installations, taking into account needs from all projects.

4.1.3 Post-installation tasks

After installing both the DOORS database server and the DOORS client start the DOORS client. This is a simple and effective check to make sure that DOORS is installed properly: if you can use the DOORS client to work with a set of sample data, the installation is done properly. It is recommended to use some sample data for this check in order not to modify any real project data.

If you experience any problems in installing DOORS, please refer to the section "Troubleshooting" of the installation guide in the **[DOORS Information Center]**. The IBM Rational support is available if you are not able to resolve your problem using that information.

Note that the certificate associated with this instance of the DOORS safety manual covers the versions of DOORS and DOORS Web Access identified on the title page of this document . If you need the certificate to be in effect for your DOORS implementation, you must use one of these versions. It is your responsibility to confirm that all users of your DOORS implementation are using certified versions of the clients.

4.1.3.1 First log in

If this is the very first access to the DOORS database you will be asked to enter a password for the user "Administrator". This is a special user account which provides the ability to do everything in DOORS. Create a user of type "Database manager" and use this account for configuring the DOORS database.

  - It is recommended to use the "Administrator" account only for tasks a database manager cannot perform.
  - If you have migrated data from some former installation and you are asked to enter the Administrator password you probably have pointed the installer to the wrong directory. Cancel the login and check for the correct location of data.

4.1.3.2 Configuration of the DOORS database server

If the database has been accessed before, log in as a user of type database manager and check the following options in the database properties.

- DOORS can keep a record of every failed login and every successful login. Configure this feature according to your preference in the section "Login History". The default is to log both unsuccessful and successful logins.
- Make sure "Allow deletion of module baselines" is not checked. This is the default setting which can only be changed by the "Administrator".
- Define the login policy:
  - restrict access to the database to certain client versions
  - demand passwords on login
  - Set the behavior on login failure
- Define the password policy, setting restrictions on valid passwords
- Set the DXL security, restricting the ability to edit DXL scripts to dedicated users, and also restricting the locations DXL scripts can be placed.

Note that the log file is never rolled over or truncated, and thus will grow until the disk is full unless managed by a person responsible.

By restricting access to the database to certain client versions

- obsolete versions with known and relevant defects can be excluded, and
- untested versions can be excluded.

Note that some new features only become available when obsolete versions of the client are excluded.

Restricting access to certain client versions may be used to make sure the certificate is in effect.

There are several advanced settings on the DOORS database server and on the DOORS client which may be set in the registry. A complete list of these options can be found on the **[DOORS Information Center]** in the sections "Configuring - Configuring the registry and using command-line switches for the Rational DOORS client", "Administering - Managing the Rational DOORS database server", and "Command-line switches for the Rational DOORS database server" resp.

Additional configuration settings may be applied to the DOORS database server using the *dbadmin* command, cf. section "Command-line switches for the Rational DOORS database server" of the **[DOORS reference manual]**. Some of this configuration is mandatory for the use of DOORS Web Access. It is highly recommended to keep records of any setting that is applied.

4.1.3.3 Configuration of the DOORS client

During the installation of the DOORS client the port and the name of the machines hosting the DOORS database server and the license server must be provided.

Errors in any of the information will result in errors that either the database or the license server is not available. You can check the entries in the registry.

Additional configuration settings may be applied to the DOORS client either by editing the registry or by providing command line switches to while starting the client, cf. section "Command-line switches for the Rational DOORS client" of the **[DOORS User Manual]**.

Examples of configuration settings applied frequently are

- setting the default open mode for modules to "read only" instead of "exclusive", or
- setting additional folders for DXL add-ins.

4.1.4 Documentation of the installation

It is recommended to document the details provided during the installation:

- name, specification (operating system, RAM, network card, ..), port, and data directory of the machine hosting the DOORS database server
- name, specification (operation system, RAM, MAC-ID or disk serial number, ..), and port of the machine hosting the Rational License Key Server
- any non-default settings described in the preceding section

If users are not managed by DOORS standard functionality but user management is switched to Rational Directory Server (RDS) instead (cf. section 4.3), document server name, port, and specification of the corresponding machine as well.

In case any integrations are set up, also record the details of each integration, including purpose, server name, ports, synchronization type (on demand or scheduled), times, intervals, or events for synchronizing, and all additional information characterizing that specific integration.

4.1.5 Responsibilities of the customer

The customer is responsible for the procurement of suitable hardware and IT infrastructure according to the system requirements. This includes machines appropriately sized, and a network providing proper parameters as latency (i.e. ping time) and band width.

The customer is responsible for the acquisition of a sufficient number of licenses.

It is out of scope of DOORS to detect and correct or compensate for hardware errors. It is the responsibility of the user to detect and correct or compensate for these types of errors by using mechanisms such as error correcting RAM, RAID or mirrored disk storage, etc. (cf. **[ECC]** and **[RAID]** for an introduction and for further reference).

4.2 Data Backup

The data of a DOORS database are stored on the file system. In order to protect against data loss, e.g. in case of hardware failure, it is recommended to define and operate some backup procedure.

There are two approaches to backup DOORS data:

- create module or project archives, and
- backup the file system.

Each approach will be described briefly, for more details cf. **[DOORS Backup]**. The definition of details like the frequency of backing up and the period how long to keep backup data, is the responsibility of the customer. General considerations concerning backup strategies apply to DOORS as well. Make sure to take backups regularly to protect the data in the DOORS database against data loss.

4.2.1 Back up using DOORS archives

DOORS archives are used to make a backup of a particular module or project. This protects against accidental data loss. For example, if a user deletes and purges a project, you can restore the project from your archive.

In order to archive a project or a module, the corresponding item must be selected in the right hand pane of the database explorer. Select File > Archive from the menu, and enter the path and name of the archive file to be created. The project and all of its contents must not be open, else the archive cannot be created.

Project and module archives should always be restored to the same database that they were created in. In the left hand pane of the database explorer select the folder you want to restore the archive in. Select File > Restore and select the archive that you want to restore. You can restore all contents of a project, or just select some items of a project that you want to restore.

It is possible to restore archives to a database different to the database of origin, but doing so results in loss of access controls previously imposed in the database of origin. This especially means that any shareable sections that might have been set up will get lost.

Make sure that you have appropriate access when archiving or restoring. Only database managers or custom users with "Create project" power are able to restore project archives, for example, and archives can only be restored to folders that you have "Create" access to.

Please be aware that archives do not update the modules or projects they were created from, but restoring archives always creates new projects or modules. So you may need to copy certain modules and recreate links if you intend to replace parts of projects only.

4.2.2 Back up the file system

Any standard file system backup tool can be used to back up the data directory of the DOORS database.

For a backup strategy of the DOORS database, consider the following facts.

- Only do full and differential backups. Do not do incremental backups. Because of the structure of the Rational DOORS database, incremental backups might capture inconsistent data.
- If you do differential backups, make sure that you do a full backup at least once a week.
- Before you start the backup, make sure that all users are logged out of Rational DOORS. Most backup tools do not back up files that are open. If files are skipped because users are accessing the database, your backup might capture inconsistent data.

You must make sure the DOORS database server service is stopped while taking the backup. DOORS does not support "hot backups" as users accessing the data while the backup is taken might cause inconsistent data.

Server downtime may be minimized by employing some snapshot technology. The Volume Shadow Copy Service (VSS) of Windows systems (cf. **[VSS]**) provides an efficient means to quickly copy the data partition. Post-processing the shadow copy while the server is running again might be used to reduce the amount of data to be saved.

In case the database server is running in a virtual environment, snapshots of the data partition may be taken, and post-processing may be applied where necessary.

4.3 User management

You only have access to a DOORS database if you have a DOORS user account. After login into the DOORS database any user is subject to the access rights schema which controls what access the user has to the data. Users may be managed by standard DOORS functionality or by employing Rational Directory Service (RDS), an application based on LDAP v3 for managing user information for all Rational lifecycle management tools.

4.3.1 Types of users

Each user account is assigned one of the following user types. The user type controls what administrative operations the user can perform.

- **Standard Users** can work with data according to the access rights they have. No administrative operations can be performed.
- **Project Managers** may perform the following operations.
  - Partition data (i.e. exchange data with other DOORS databases)
  - Archive data (i.e. create a backup of a project or module)
  - Create and manage user groups
- **Database Managers** in addition to the powers of a project manager may perform the following operations.

    - Create projects
    - Create and manage users
    - Manage the database (fundamental changes to database properties)
- **Custom Users** can be assigned any combination of the powers a database manager may perform.

In addition there is one super user account **Administrator** for each DOORS database. This account is not subject to the access rights schema, so it is always allowed to do everything in all the DOORS database. Consequently its password must be kept safe.

Some actions, like e.g. updating the database to a new major version, or switching to RDS user management, can only performed using the Administrator account. This account should only be used for such actions.

Most users of DOORS will be standard users, there will be several Project Managers and Custom Users, and few Database Managers.

It is a best practice to establish a role concept, defining what user groups support the development process, and what access rights each group is granted.

DOORS supports the access rights as defined subsequently.

R - read access, the user can see the item

M- modify access, the user can modify the item

C - create access, the user can create items at the next level in hierarchy, e.g. create a module in a folder

D - delete access, the user can delete, undelete and purge the item

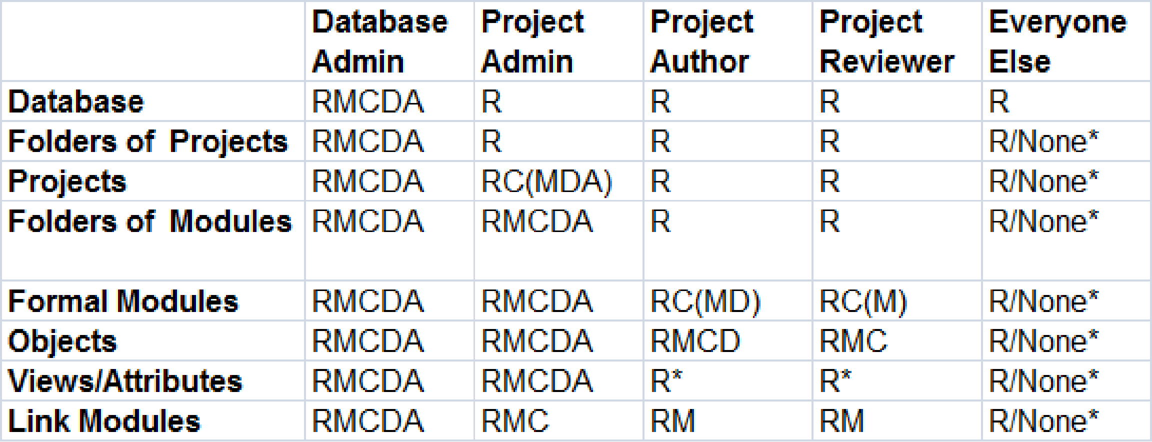
A - admin access, the user can change access setting for the item

If a user has access to an item as a member of several groups, the most permissive rights apply, that is the union of all rights granted.

Special attention must be paid to the group "Everyone" or "Everyone Else". This group is dynamically assigned to all DOORS users that are not mentioned explicitly or as members of other groups. Also note that this group may contain different sets of users at various items. It is a good practice to either grant "Read Only" or "None" to the group "Everyone Else", depending on how secret project information is considered.

At the database, read access cannot be removed from any user. If you want to prevent a user from logging in, that user account can be deactivated.

An example for an access rights schema is given in the following figure.



	Database Admin	Project Admin	Project Author	Project Reviewer	Everyone Else
Database	RMCDA	R	R	R	R
Folders of Projects	RMCDA	R	R	R	R/None*
Projects	RMCDA	RC(MDA)	R	R	R/None*
Folders of Modules	RMCDA	RMCDA	R	R	R/None*
Formal Modules	RMCDA	RMCDA	RC(MD)	RC(M)	R/None*
Objects	RMCDA	RMCDA	RMCD	RMC	R/None*
Views/Attributes	RMCDA	RMCDA	R*	R*	R/None*
Link Modules	RMCDA	RMC	RM	RM	R/None*

Figure 10: Example of access rights for several organizational roles

The asterisk (*) in the above figure indicates that the appropriate access rights settings depend on the security needs. In specific situations more restrictive or more permissive settings may be the proper choice.

4.3.2 Native DOORS user management

By default DOORS users are managed by standard DOORS functionality, user records being maintained in the DOORS database.

For each user certain information must be provided, like the user name, the user type, and password settings. In addition optional details like email address, telephone number, and location may be provided.

It is strongly recommended to manage access rights using groups rather than using individual users. A user group is characterized by its name. Groups may have any number of users as members, and each user may be a member of any number of groups.

Users can only be created and managed by database managers or custom users having the power "Create users" whereas groups can be created and managed by database managers, project managers, and custom users having the power "Create groups". Users must change their password at the first login, whether or not a password was set on creation of the account. By default the initial password is blank.

A user archive is a backup copy of all information about users and groups. Database managers, project managers, and custom users having the power "Create archive" may create user archives, but only the Administrator can restore a user archive. By restoring a user archive the current user information is replaced with the information in the archive, which includes the Administrator password.

4.3.3 Managing users with the Rational Directory Server

DOORS supports user authentication and administration by the Rational Directory Server (RDS), which is based on LDAP v3. RDS is used to manage user information for Rational lifecycle tools in one place. Only the Administrator can configure DOORS to use RDS.

The use of RDS is not covered by the certificate.

4.4 Information architecture

DOORS provides the flexibility to support a wide variety of processes. This entails the necessity to be set up accordingly in order to provide support for the process and guidance for the DOORS users. A good practice is to define and to document a common DOORS information architecture for use in all projects of a DOORS database.

4.4.1 Elements of a DOORS information architecture

A DOORS information architecture specifies what projects, folders, and formal modules exist. It also defines what relations between formal modules are needed and how this schema is supported by link modules. Furthermore the information architecture determines what attribute types, attributes, and views exists in each formal module. The access right schema, based on a role concept, is part of the information architecture, too.

The DOORS information architecture typically is the outcome of a workshop, reflecting both the processes of the customer and best practices for using DOORS. It is the responsibility of the customer to appoint some experienced DOORS user with the task to maintain the information architecture and to ensure that the projects are actually consistent with the information architecture.

One of the major benefits of employing an appropriate and uniform information architecture in all projects is the option of generating reports on various projects of a DOORS database.

4.4.2 Crucial subjects

From experience it is worth emphasizing the following topics for thorough consideration in the DOORS information architecture.

- A linking schema must be defined and enforced. A linking schema defines what link modules to use for documenting relations between objects in the DOORS database. The enforcement prevents the creation of links not in accordance with the link schema. A consistent linking schema facilitates traceability analysis.
- Access right must be restricted properly. Restricted access rights constitute an effective means to control the development of a DOORS database. The access rights on link modules deserves special attention in order to protect the linking schema from unintended changes.

4.4.3 A sample information architecture

It is mandatory to document the information architecture as a basis for future development. The documentation may be a textual description of the structure, or it may be any other suitable format, as e.g. a UML diagram, cf. the following figures.

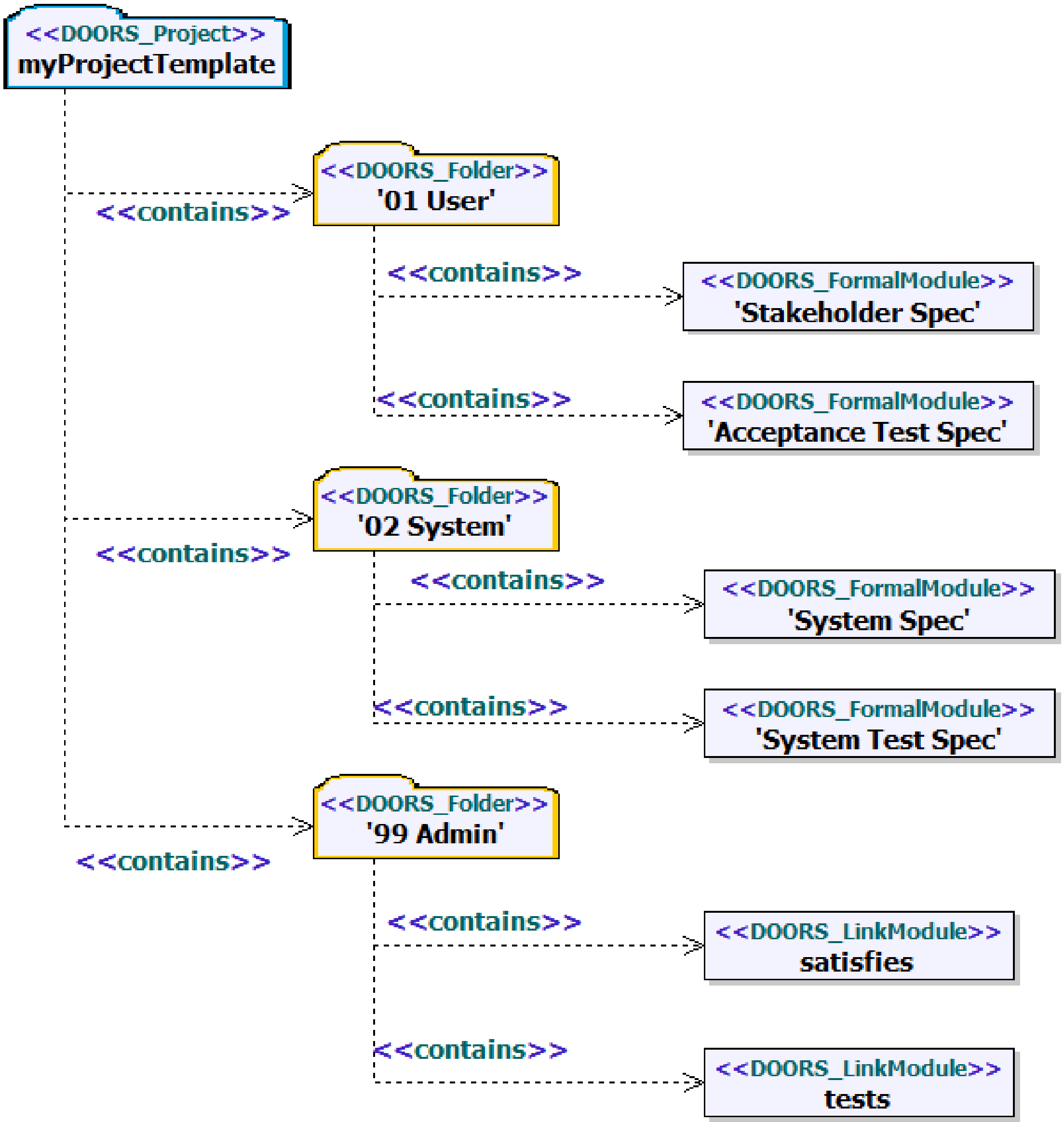


Figure 11: Sample UML diagram describing the database structure

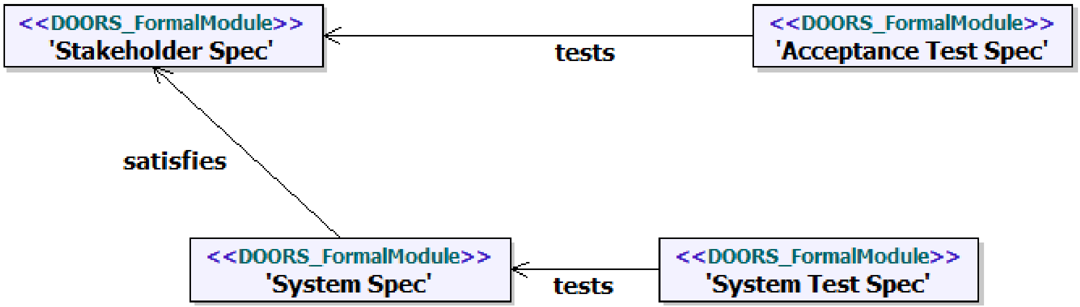


Figure 12: Sample UML diagram defining the linking schema

The documentation should provide justification for decisions, and it should capture additional information like e.g. the role responsible for maintaining certain attributes.

4.4.4 Recommended attributes

This section lists the attributes that are recommended to document the application of the ISO standard. As ISO 26262 is the adaption of IEC 61508 (cf. the introduction of any part of the ISO), the attributes listed in this section are recommended for the ICE 61508 as well. Attributes that support the development process but are not related to safety are not included in this section. A complete list of attributes cannot be given here as this list reflects the development process. So this list must be determined during an information architecture workshop for each customer individually.

For each attribute we provide a description, define what attribute type to use, and suggest who is responsible for maintaining the attribute.

The history of DOORS keeps records about changes, including the user, the time and date, the attribute changed, and the old and new values of the attribute. This information can be extracted from the history records, so no separate attributes are needed to track changes.

4.4.4.1 Attribute "ASIL"

The attribute "ASIL", required by ISO 26262-8, 6.4.2.5 c), classifies the hazardous events for which safety requirements are determined according to ISO 26262-3, 7.4.4. If one safety requirement is defined for several hazardous events, the highest ASIL is assigned to the safety requirement. This attribute should be using an enumeration as attribute type. The values of the enumeration are implied by ISO 26262-3, table 4.

- UNDETERMINED - ASIL cannot be determined because parameters are missing
- QM - quality management, no requirement to comply with ISO 26262
- A - least stringent level or lowest level
- B
- C
- D - most stringent level or highest level

The adequate implementation of the "ASIL" attribute depends on its precise use. Consider the following three examples:

1. The requirement is a safety goal, i.e. a top level safety requirement, and the ASIL must be determined according to ISO 26262-3, table 4, based on the classification of controllability, probability of exposure, and severity of hazardous events (cf. subsequent section 4.4.5). The determination of the ASIL can be automated by a DXL attribute, implementing the decisions described by table 4.

2. The requirement is a safety requirement, derived from safety goals or other safety requirements, and its ASIL is determined according to ISO 26262-3, 7.4.4.3 and 7.4.4.4 as the highest ASIL of all requirements or safety goals from which the present requirement is derived. If DOORS standard links document this kind of relationship, the determination of the ASIL can be automated by a DXL attribute, implementing the specification of sections ISO 26262-3, 7.4.4.3 and 7.4.4.4.

3. The requirement is not derived from other requirements, and its ASIL has been determined already, so no further action is required. This case exists if the engineering group receives the requirement from another organization that defined the ASIL such as when manufacturers give requirements with ASIL designations to suppliers.

4.4.4.2 Attribute "Object Type"

ISO 26262-8, 6.4.2.1 requires that safety requirements shall be unambiguously identifiable. This can be achieved in DOORS objects by using an attribute "Object Type", which is using an enumeration as attribute type. As the values of the enumeration cannot be deduced from the ISO, the list given below must be considered as an example. Additional values may be included if needed to provide better support to the development process.

- Safety Goal - a top level safety requirement
- Safety Requirement - a requirement addressing safety, derived from safety goals or other safety requirements
- Requirement - any requirement not addressing safety
- Information - any contents that is not a requirement, used to provide context to requirements
- Heading - section headings, defining the structure of the document
- Not Set - the default value for new objects

The "Object Type" must be defined by the author of the requirement, and it must be subject to review. The default value "Not Set" must be changed to indicate the appropriate type of object.

The "Object Type" provides redundant information that can be used to identify if the ASIL has been determined for all relevant requirements.

4.4.4.3 Attribute "Object Identifier"

ISO 26262-8, 6.4.2.5 a) requires that safety requirements shall have a unique identification remaining unchanged. The DOORS "Object Identifier" meets this requirement. This attribute is created and maintained by DOORS, so no user action is necessary.

4.4.4.4 Attribute "Object Text"

In DOORS, the text of requirements typically is captured in the "Object Text" attribute. This attribute is created by DOORS, so it need not be explicitly defined.

The author of the requirement must enter the text of the requirement into the "Object Text"

4.4.4.5 Attribute "Requirement Status"

ISO 26262-8, 6.4.2.5 b) requires that safety requirements have a status. As several status values may be used to characterize requirements, e.g. a test status and a realization status, it is best practice to call this attribute "Requirement Status". This attribute should be using an enumeration as attribute type, reflecting the development process. The list given below is an example, additional values may be included if needed to provide better support to the development process.

- new
- proposed
- assumed
- accepted
- reviewed
- rejected

The default value "new" must be changed by the author to "proposed" as the requirement is considered to comply with all criteria, including those listed in ISO 26262-8, 6.4.2.4. The development process must define responsibilities for subsequently changing the status value.

4.4.4.6 Attribute "Review Comment"

In order to keep records of comments made during the review process, several options are viable. The easiest is to define an attribute called "Review Comment", which is of type text. Reviewers capture any information in that attribute.

4.4.4.7 Attribute "Review Status"

In order to make sure that all requirements have been reviewed, reviewers keep records of their actions using an attribute "Review Status". This guards against the possibility of assuming a requirement was reviewed and the reviewer had no comments when it was not reviewed. This attribute should be using an enumeration as attribute type, reflecting the review process. The list given below is an example which must be adapted to the customer's review process.

- not reviewed
- in review
- OK
- ERR

The default value "not reviewed" must be changed by the reviewer to one of the other values, depending on the assessment results.

4.4.5 More useful attributes

In contrast to the attributes described in the previous section, which are recommended at all levels of the development, the attributes specified below are useful only in the corresponding module if the ASIL is determined in DOORS according to ISO 26262-3, table 4. If the ASIL is inherited from the level above or if the ASIL is determined outside of DOORS, the corresponding modules can do without these attributes.

4.4.5.1 Attribute "Controllability Class"

The attribute "Controllability Class" classifies the controllability of hazardous events according to ISO 26262-3, 7.4.3.7 and 7.3.4.8. This attribute should be using an enumeration as attribute type. The values of the enumeration are implied by ISO 26262-3, table 3.

- C0 - controllable in general

- C1 - simply controllable
- C2 - normally controllable
- C3 - difficult to control or uncontrollable
- N/A - not applicable
- Not Set - default value

If class C0 is assigned, the determination of an ASIL is not required. N/A is assigned to requirements that do not address safety.

The "Controllability Class" must be defined by the author of the requirement, and it must be subject to review. The default value "Not Set" must be changed by the author to indicate the appropriate classification.

4.4.5.2 Attribute "Probability Class"

The attribute "Probability Class" classifies the probability of exposure to a hazardous event according to ISO 26262-3, 7.4.3.4 - 7.4.3.6. This attribute should be using an enumeration as attribute type. The values of the enumeration are implied by ISO 26262-3, table 2.

- E0 - incredible
- E1 - very low probability
- E2 - low probability
- E3 - medium probability
- E4 - high probability
- N/A - not applicable
- Not Set - default value

If class E0 is assigned, the determination of an ASIL is not required. N/A is assigned to requirements that do not address safety.

The "Probability Class" must be defined by the author of the requirement, and it must be subject to review. The default value "Not Set" must be changed by the author to indicate the appropriate classification.

4.4.5.3 Attribute "Severity Class"

The attribute "Severity Class" classifies the severity of potential harm of a hazardous event according to ISO 26262-3, 7.4.3.2 and 7.4.3.3. This attribute should be using an enumeration as attribute type. The values of the enumeration are implied by ISO 26262-3, table 1.

- S0 - no injuries
- S1 - light and moderate injuries
- S2 - severe and life-threatening injuries (survival probable)
- S3 - life-threatening injuries (survival uncertain), fatal injuries
- N/A - not applicable

- Not Set - default value

If class S0 is assigned, the determination of an ASIL is not required. N/A is assigned to requirements that do not address safety.

The "Severity Class" must be defined by the author of the requirement, and it must be subject to review. The default value "Not Set" must be changed by the author to indicate the appropriate classification.

4.4.6 Extended information architecture

If DOORS is not used as a stand-alone tool but is integrating with other tools, the corresponding interfaces must be accounted for in the information architecture as well. As there are numerous tools integrating with DOORS, no complete guideline can be given here. The general procedure is to analyze the process upon which the integration is based in first place, and then to map its items to DOORS items, and to items of the integrating tool as well.

Domains, that either can be handled in DOORS or that need integrations in place, comprise

- Risk management / risk analysis
- Hazard management
- Design management - typically integrating with some modeling tool
- Quality management - typically integrating with some test management tool
- Change management - typically integrating with some change management tool

4.5 Sample Workflow

There are a wide possibility of use cases in any user environment and it is not possible to document them all. However, there is a general pattern for DOORS workflow that should be applied to all projects that use DOORS to capture and manage safety critical information.

This process assumes that DOORS has been installed and set up correctly using the guidance given in previous sections of this chapter, modules and data architecture have been established, and user access rights defined.

Section 6.3 of this document gives guidance on how to conduct a formal review and a sample Formal Requirements Review checklist.

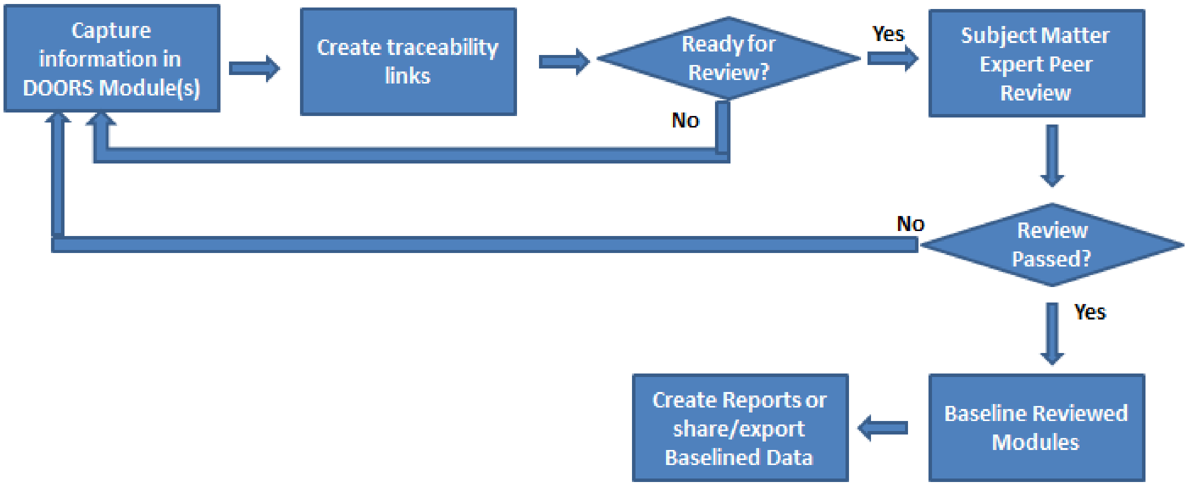

Capture information in DOORS Module(s)
Create traceability links
Ready for Review?
Yes
No
Subject Matter Expert Peer Review
Review Passed?
No
Yes
Baseline Reviewed Modules
Create Reports or share/export Baselined Data

Figure 13- DOORS Usage Reference Workflow

5 Features of DOORS

This section describes the features of DOORS that the certificate applies to. The features listed here correspond to the sections of the **[DOORS user manual]**, where additional information on features or actions supported by features may be found.

In this section the features of DOORS are organized according to the following topics, differing from the DOORS manual.

- Features at database level
- Features for documents
- Features for displaying information
- Features for links
- Features for data exchange
- Additional features
- Features for customizing DOORS

Each subsection provides a description of the corresponding **feature** and lists the actions that are supported by the feature. If appropriate, the feature is differentiated from related features. Features at database level

This section describes the features used to navigate the database to locate projects, folders, and modules.

5.1.1 Database Explorer

The database explorer allows DOORS users to navigate through the projects, folders, and modules in the database.

The feature "Database Explorer" supports the following actions.

- Cutting, copying, and pasting in the database explorer
- Deleting, undeleting, and purging
- Copying the structure of an existing module
- Entering paths
- Using favorites

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.1.2 Finding data in a DOORS database

Searches can either be initiated from the DOORS database explorer, or from within individual modules. As a result of a search from the database explorer a list of matching projects, folders or modules is displayed. Searching within a module sets the focus to the next matching object.

Within modules, usually filters are used instead of searching.

The feature "Finding data in a DOORS database" supports the following actions.

- Searching the database
- Searching and replacing text in modules
- Regular expressions for searching and filtering modules
- Going to a particular object in a formal module

Note that using filters is not covered by the present feature, but is described by the feature "Using Views" in section 5.3.2.

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.1.3 Projects and folders

Projects and folders are used to organize and structure the data in the database. They can be created anywhere in the database hierarchy. Compared to folders, projects have additional functions used for administration at database level.

The feature "Projects and folders" support the following actions.

- Access rights for projects and folders
- Locks
- Creating projects
- Creating folders
- Editing project and folder properties
- Converting a project or a folder

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.2 Features for documents

Documents are called "formal modules" in DOORS. Formal modules are composed of Objects, which may contain embedded objects, and they define the scope for attribute definitions. Changes to module contents are recorded in the module history.

5.2.1 Formal modules

Formal modules contain information in objects that are defined by their attributes.

The feature "Formal modules" supports the following actions.

- Creating modules
- Copying the structure of an existing module
- Managing access rights for formal modules
- Managing open modules
- Showing module statistics
- Selecting toolbars to show in modules

Editing contents of formal modules is not covered by this feature, but is described by the feature "Editing formal modules". The access to formal modules is controlled by the access rights it is assigned. Access rights are not covered by this feature, but are described by the feature "Access Rights".

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.2.2 Editing formal modules

DOORS formal modules consist of a hierarchy of objects, which in turn contain information, as e.g. headings, requirements, and background information.

Unlike the feature "Objects" this feature does not address single objects, but describes how the module supports managing a number of objects.

The feature "Editing formal modules" supports the following actions.

- Selecting an edit mode
- Creating editable sections in modules
- Locking and unlocking editable sections
- Using the object properties sheet to edit objects and attributes
- Editing in the module window
- Using forms to edit objects and attributes
- Creating and editing tables
- Inserting modules as tables
- Inserting symbols
- Undoing edits
- Checking spelling

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.2.3 Objects

An object is a database record. Objects are members of a formal module. They store their information in attributes that are defined within the module.

The feature "Objects" supports the following actions.

- Creating objects
- Selecting objects
- Copying objects
- Dragging objects
- How copy and move affect links
- Manipulating objects
- Deleting, undeleting, and purging objects

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.2.4 Managing Attributes

In DOORS, information about modules and objects is stored in attributes. This feature deals with the administration of attributes. Management of attribute values is addressed by the features "Editing formal modules" and "Objects".

The feature "Attributes" supports the following actions.

- Managing attribute types
    - Creating attribute types
    - Editing attribute types

    - Deleting attribute types
- Managing attribute definitions
    - System attributes
    - Locales for textual attributes
    - Creating an attribute definition
    - DXL attributes and layout DXL columns
    - Editing attribute definitions
    - Deleting attribute definitions
- Reusing attribute types and attribute definitions
- Copying attribute values
- Measuring the frequency of attribute values

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.2.5 OLE objects and pictures

DOORS objects may contain OLE objects. OLE objects represent data managed by a separate application. DOORS may display the contents of that information, or it may display an icon.

DOORS can also represent OLE objects as pictures objects, which unlike OLE objects cannot be edited.

The feature "OLE objects and pictures" supports the following actions.

- Editing OLE objects
- Inserting OLE objects
- Inserting pictures
- Editing OLE object properties
- Recording history for OLE objects
- Handling unregistered OLE objects

As OLE objects are not modified by DOORS but by the separate application (e.g. Excel tables are edited with Microsoft Excel, pictures are edited with some graphics program), editing OLE objects is out of scope. This feature covers the handling of OLE data by DOORS, e.g. inserting and maintaining the link information.

During a formal review all OLE object should be manually validated to check for correctness. This will detect problems such as a link made to information referenced on a local, not networked, storage location.

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.2.6 Tracking changes to DOORS data

DOORS records changes to objects and modules in a history. Based on history records and links users are supported to track changes to data, and to manage the impact that changes have to other items in the database.

The feature "Tracking changes to DOORS data" supports the following actions.

- Change bars
- Module history
- Redline markup in DOORS
- Suspect links and changed objects

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.2.7 Discussions

Discussions allow developers to exchange views about the content of a module or the content of an object in a module without needing modify access to the module or object. Discussions are presented as part of the properties of the object or module.

Please note that if you archive and restore a module, discussions about the module or objects in the module are not restored. Therefore it is strongly recommended to use discussions for informal exchange of information only, but not for formal reviews. If this recommendation is followed, this feature in general will not be safety critical, supporting the communication on information captured in DOORS objects or modules.

The feature "Discussions" supports the following actions.

- Creating discussions
- Controlling access for discussions in a module
- Viewing discussions and adding comments
- Closing, reopening, and deleting discussions

This set of features has a generic Tool Confidence Level (TCL) 3 because discussions are not restored during the archive/restore process. Discussions can be used informally but should not be used to store information required for audit or other permanent usage.

5.3 Features for displaying information

DOORS holds the contents of modules in a database. Views are used as a means to control the way the data is presented to the users. Filters and sorting criteria may be saved as part of a view.

5.3.1 Controlling the data that is displayed in formal modules

DOORS allows users to define a layout for module data that is most suitable for a certain task. The layout includes columns, filters, sorts, and what levels of the hierarchy to display. A layout intended to be used again is saved as a view that can be accessed every time the module is open.

By default after opening a formal module the Standard view is displayed, showing the module explorer, the ID column, and the main column.

Users may save additional views containing e.g. additional columns, applied filters or sorts, changed size of the module window. Users may select views from a list. Each view may be set as default, so users are presented this view instead of the Standard view.

Filters let the user control the data that is displayed in a module by including or excluding objects according to simple or combined conditions. A filter may or may not be saved as part of a view.

The features "Controlling the data that is displayed in formal modules" supports the following actions.

- Defining filters
- Regular expressions for searching and filtering modules
- Defining sorts
- Selecting the level of objects to show in modules
- Selecting the attribute to show in tables
- Inserting, editing, and removing columns
- Inserting graph columns
- Inserting icon columns
- Saving views
- Viewing, editing, and deleting views

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.4 Features for links

Traceability in DOORS is created using links. DOORS offers several features supporting the creation and the analysis of traceability. DOORS standard links establish bidirectional traceability within a single DOORS database. Link modules and linksets help setting up a schema forcing DOORS users to create links as intended. External links and DOORS URLs support navigation from and to DOORS, while collaboration links establish a relationship between DOORS information and information held in other Rational or non-Rational tools that have been set up correspondingly, such as e.g. design artifacts, test artifacts, or change management artifacts.

5.4.1 Standard links, external links, and collaboration links

Links are used to associate several pieces of information. DOORS supports several types of links depending on the nature of information.

- Standard links associate requirements in the same DOORS database with one another.
- External links associate requirements in a Rational DOORS database with an entity or resource that is outside the current DOORS database.
- Collaboration links associate requirements with artifacts in a server that has been set up using the Rational solution for Collaborative Lifecycle Management (CLM).

While external links provide a means of navigation from a DOORS database to resources outside of that database, the feature "DOORS URLs" allow users to navigate from resources outside of DOORS to a specific item in the DOORS database.

As external links and collaboration links leave the responsibility of DOORS, their usage is not covered by this certificate. If users want to make use of external links or collaboration links, it is their responsibility to ensure qualified usage of the application these links refer to.

If external links are used, it is the responsibility of the user to ensure that the items pointed to by external links exist and are maintained. If the external links are pointing to some other application, like Microsoft Work documents, it is also the responsibility of the user to make sure that this usage is covered by their process. If a user navigates to an external URL that is not available or not valid, then they will get an error message stating that the browser cannot display the webpage, or it cannot find the file. The user then needs to determine if the link was set correctly (if not previously validated and baselined), or if the target information was moved incorrectly and has to be restored, or if the broken link has to be fixed by other means.

The feature "Standard links, external links, and collaboration links" supports the following actions.

- Creating links
- Creating multiple links
- Creating external links
- Creating collaboration links
- Creating links by attribute
- Editing links
- Deleting links from multiple objects

This feature describes how to maintain individual links. Configuration of a link schema and analyzing traceability are not covered by this feature, but by the features "Link modules, linksets, and linkset pairings" and "Links and traceability", respectively.

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.4.2 Link modules, linksets, and linkset pairings

For standard links, DOORS supports setting up and enforcing a schema for linking.

Link modules store information about standard Rational DOORS links. For each type of link, e.g. "satisfies" linking requirements and "tests" linking test cases to requirements, a different link module is used. Within each link module, the information is subdivided into linksets, each representing relationships from one particular module to another.

Each formal module can be configured to specify what target modules and what type of link may be used for establishing links.

The feature "Standard links, external links, and collaboration links" deals with using links, whereas the present feature "Link modules, linksets, and linkset pairings" is concerned with administrative aspects of standard DOORS links.

The feature "Link modules, linksets, and linkset pairings" supports the following actions.

- Creating link modules
- Defining default link modules
- Editing link modules
- Viewing, creating, and deleting linksets
- Creating link attributes
- Creating linkset pairings
- Managing access rights for link modules

This feature describes the configuration of a link schema. Maintaining or analyzing traceability is not covered by this feature, but is described by the features "Standard links, external links, and collaboration links" and "Links and traceability", respectively.

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.4.3 Links and traceability

Links between objects provide the basis for several types of analysis, supporting users in assessing requirements coverage and completeness of the specification. Traceability also helps in managing change by quickly identifying what impact a proposed change has on the project.

The feature "Standard links, external links, and collaboration links" deals with using links, whereas the present feature "Links and traceability" is concerned with navigating and analyzing links.

The feature "Links and traceability" supports the following actions.

- Analyzing links
- Adding a traceability column
- Analyzing links using the traceability explorer

This feature describes the analysis of traceability. Maintaining traceability and configuration of a link schema are not covered by this feature, but they are described by the features "Standard links, external links, and collaboration links" and "Link modules, linksets, and linkset pairings", respectively.

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.4.4 DOORS URLs

Each item of the DOORS database, including the database, projects, folders, modules, baselines, and objects has a unique identifier, which is expressed as a URL. The URL of each item is stored in its properties sheet, and can be copied from there. The DOORS URL is a means to navigate from other applications, as e.g. a Microsoft Word document, to the specified DOORS item.

If DOORS URLs are used to navigate from some other application to DOORS items, it is the responsibility of the user to make sure that this usage is covered by their process.

During formal reviews, URLs, like OLE objects, should be exercised to ensure they point to reachable data (i.e., to objects that exist in a network storage location).

The feature "DOORS URLs" supports the following action.

- Following DOORS URLs

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.5 Features for data exchange

There are several ways to exchange data between DOORS and other tools, or to extract data from DOORS. Each of these features will be described in the subsequent sections.

5.5.1 ReqIF

The Requirements Interchange Format (ReqIF, also called RIF up to 2010) is used to exchange requirement information between DOORS databases.

DOORS data can be send for editing to another DOORS database. After the data has been edited, it is returned to the originating DOORS database, and, if appropriate, merged with the original data. Returning the data and merging it is a two-step process.

Please note that the DOORS menu items as well as the DOORS reference manual use "RIF" for Requirements Interchange Format for the versions analyzed

The feature "ReqIF" supports the following actions.

- Sending and receiving RIF files
- RIF locks on local data
- Creating RIF definitions

- Editing RIF definitions
- Deleting RIF definitions
- Exporting RIF packages
- Importing RIF packages
- Merging RIF packages
- Recovering RIF locks

Your process should include checking that the data was received successfully with no data corruption. This could be accomplished by validating that the exported count of data equals what was received. As another alternative, both parties could use a check-sum tool on the exported data and verify that the same result was obtained on both ends.

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.5.2 Data partitions

Data partitions are a means to exchange data with users of a different DOORS database. In the away database the data partitioned out may either be viewed or edited and then returned to the home database. Data that may be edited by the users in the away database is locked in the home database until it is returned and rejoined, or recovered.

The feature "Data partitions" supports the following actions.

- Sending and receiving partition files
- Creating partition definitions
- Editing partition definitions
- Deleting partition definitions
- Exporting partition definitions
- Importing a partition
- Showing exported and imported partitions
- Adding data to imported partitions
- Creating a sync file from an imported partition
- Synchronizing an exported partition
- Returning an imported partition
- Rejoining a partition
- Recovering a partition

Your process should include checking that the data was received successfully with no data corruption. This could be accomplished by validating that the exported count of data equals what was received. As another alternative, both parties could use a check-sum tool on the exported data and check that the same result was obtained on both ends.

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.5.3 Exporting data from formal modules

DOORS exporters extract data from formal modules to other applications or formats. The contents of the export are generally controlled by selecting a view for export.

The feature "Exporting data from formal modules" supports the following actions.

- Exporting to Microsoft Word
- Exporting to Microsoft Excel
- Exporting to Microsoft Outlook
- Exporting to PowerPoint
- Exporting to HTML
- Exporting to plain text
- Exporting to RTF
- Exporting to spreadsheet
- Exporting to FrameMaker

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.5.4 Exporting data from formal modules using Rational Reporting for Document Generation

Rational Reporting for Document Generation extracts data from DOORS modules into templates, defining the layout of the report. The report may be generated in Microsoft Word format, in PDF format or in HTML format.

The feature "Exporting data from formal modules using Rational Reporting for Document Generation" supports the following actions.

- Configuring the Document Generation defaults for the module
- Exporting to Microsoft Word using Rational Reporting for Document Generation
- Exporting to PDF using Rational Reporting for Document Generation
- Exporting to HTML using Rational Reporting for Document Generation

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.5.5 Printing and reports

The contents of formal modules can be sent to a printer. DOORS uses page setups to control the layout of the printout. DOORS always prints views, i.e. those objects and those columns that are displayed. A report combines a page setup and a view, thus providing a means to print from the database explorer without opening the formal module first.

The feature "Printing and reports" supports the following actions.

- Creating, editing, and applying page setups
- Printing modules
- Creating and managing reports for printing

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.5.6 Importing data into formal modules

DOORS importers populate formal modules with data from other applications.

The feature "Importing data into formal modules" supports the following actions.

- Exporting from Microsoft Word to DOORS
- Editing imported style information from Word
- Importing plain text files
- Importing rich text format files
- Importing spreadsheets
- Importing FrameMaker files

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.6 *Additional features*

DOORS provides several additional features which do not group to any of the preceding topics.

5.6.1 Authenticate

DOORS provides specific access to the data to users. So each user has to authenticate providing credentials before achieving access to DOORS data. Credentials may be supplied directly by entering user name and password, or they may be read from a smart card like CAC (Common Access Card).

Only registered users can authenticate at a DOORS database. Every user who has authenticated successfully is subject to the DOORS access rights which control what they are allowed to do in the DOORS database.

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.6.2 Access rights

Access rights control what users can do to each item in a DOORS database. Access rights can be assigned to individual users or to groups of users.

The feature "Access rights" supports the following actions.

- DOORS groups
- Access rights and inheritance
- Propagating additional access rights with create access

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. This data is not viewed or used on a common basis or reviewed as part of a peer review. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.6.3 Baselines

In DOORS, a baseline is a read-only version of a module. The baseline captures the status of the module at a certain time and preserves it until the module is deleted.

Baseline includes the module history:

- Information about all the attribute definitions and types that have been created, deleted, or edited since the most recent baseline.
- Information about all the objects that have been created, deleted, or edited since the most recent baseline.
- Information about every module session (every time the module has been opened) since it was first created.

It is recommended to create baselines regularly to improve performance. The history file is removed from the module, and a file that contains the baseline, together with all the history information is created. So the history information remains in the system, it can be accessed from the baseline.

Several modules that form a unit for project planning and project management purposes can be combined in a baseline set definition. Baseline sets can be created for each phase of the project from this baseline set definition. A baseline set presents a snapshot of the data in the modules and the links that existed between them when the baseline set was closed.

The feature "Baselines" supports the following actions.

- Creating baselines at project level (baseline sets)
  - Creating baseline set definitions
  - Copying baseline set definitions
  - Removing baseline set definitions
  - Creating baseline sets
  - Closing baseline sets
- Creating baselines
- Copying baselines
- Opening baselines
- Comparing baselines
- Deleting baselines
- Electronic signature

This set of features has a generic Tool Confidence Level (TCL) 3 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit. The output of this data is typically not reviewed. The TÜV SÜD independently audited the test suites for these features for completeness and robustness and the test suite results for the covered DOORS releases so these features can be used with confidence on projects up to ASIL “D” or SIL “3” (per ISO 26262:2011-8 clause 11.4.7).

5.6.4 Keyboard shortcuts

Keyboard shortcuts allow DOORS to be operated avoiding the mouse.

The feature "Keyboard shortcuts" supports the following actions.

- Using control keys
- Applying rich text formatting
- Using function keys
- Using keypad keys
- Using navigation keys

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.7 Features for customizing DOORS

DOORS allows users to extend its functionality by providing the scripting language DXL, the DOORS Extension Language.

DXL programs written by the user are considered as separate tools, for which the author can follow a procedure to qualify that functionality like the one described for DOORS in this report. In that case, the analysis is performed for a tool chain, consisting of DOORS and the DXL tool.

This section covers the core DXL command as provided by DOORS. Programs written using the DXL language are outside the scope of this safety manual.

5.7.1 Using DXL (the DOORS Extension Language)

The DOORS eXtension Language (DXL) is a scripting language having a syntax like C and C++. It can be used to control and extend DOORS functions. Typical uses of DXL include

- Automate routine or complex tasks, such as calculating attribute values.
- Respond to events by triggering custom programs.
- Add your own options to DOORS menus.

A full description of DXL is given in the **[DXL Reference Manual]**, which is available at the "Featured Documents" section of the reference manual as a PDF and from the DOORS Help menu.

The feature "Using DXL" supports the following actions.

- DXL encryption
- Setting up DXL security
- The DXL library
- Converting layout DXL to attribute DXL

Any DXL code written by the user or a third party is not covered by the certificate. It is the responsibility of the user to make sure that any DXL add-on used in safety critical context meets the requirements of the standards, including analyzing use case coverage. DXL sample programs delivered as means to illustrate the usage of DXL are explicitly included in the preceding statement.

DXL sample programs are provided as is. Each user needs to verify them in their own environment, proving they are useful and correct for their intended purpose.

After adding a new DXL add-on the user should check that the corresponding menu item works as intended. This is a one-time action to be done be the staff responsible for providing add-ons. It is highly recommended to do the check on some sample, not live, data.

This set of features has a generic Tool Confidence Level (TCL) 1 based on analysis done by IBM and confirmed by the TÜV SÜD in an independent audit.

5.8 Summary of Critical Features of DOORS

These features described above are designated as Tool Impact 2 (TI2 per ISO 26262:2011-8 clause 11.4.3.2) in a table summarizing the generic Tool Confidence Level for each feature.

Section #	Feature Name	Generic TCL
5.1	Features at database level	
5.1.1	Database Explorer	TCL1
5.1.2	Finding data in a DOORS database	TCL1
5.1.3	Projects and folders	TCL3
5.2	Features for documents	
5.2.1	Formal modules	TCL3
5.2.2	Editing formal modules	TCL1
5.2.3	Objects	TCL1
5.2.4	Managing Attributes	TCL3
5.2.5	OLE objects and pictures	TCL1
5.2.6	Tracking changes to DOORS data	TCL3
5.2.7	Discussions	TCL3
5.3	Features for displaying information	
5.3.1	Controlling the data that is displayed in formal modules	TCL1
5.4	Features for links	
5.4.1	Standard links, external links, and collaboration links	TCL1
5.4.2	Link modules, linksets, and linkset pairings	TCL3
5.4.3	Links and traceability	TCL1
5.4.4	DOORS URLs	TCL1
5.5	Features for data exchange	
5.5.1	ReqIF	TCL1
5.5.2	Data partitions	TCL1

5.5.3	Exporting data from formal modules	TCL3
5.5.4	Exporting data from formal modules using Rational Reporting for Document Generation	TCL3
5.5.5	Printing and reports	TCL3
5.5.6	Importing data into formal modules	TCL3
5.6	Additional features	
5.6.1	Authenticate	TCL1
5.6.2	Access rights	TCL3
5.6.3	Baselines	TCL3
5.6.4	Keyboard shortcuts	TCL1
5.7	Features for customizing DOORS	
5.7.1	Using DXL (the DOORS Extension Language)	TCL1

5.9 Non-critical Features

The features described in this section will not introduce data errors and are classified as Tool Impact 1 (TI1 per ISO 26262:2011-8 clause 11.4.3.2).

Some features of DOORS cannot possibly modify safety critical data. So it is safe to use them in safety critical context.

5.9.1 Messages

Database managers can broadcast a message to all the users that are connected to the DOORS database server. For example, if the database server needs to be restarted, a message can warn all the connected users to save their work and log out of the database.

Standard users can communicate with other DOORS users by means of messages, e.g. to request the lock for a module or for a shareable section.

5.9.2 Checking database integrity

DOORS relies on the underlying integrity of the network to complete certain database operations. If contact between the client and the server is lost while DOORS is saving data, the integrity of the database can be affected.

The feature "Checking database integrity" supports the following actions.

- Identifying and repairing problems with the integrity of the database due to network, disc, or other issues.
- Comparing and restoring data from the module history

When planning to use the database integrity check, please be aware of the following.

- The database integrity check just collects conspicuous data. It does not add any items to projects without explicit user interaction.
- The database integrity check can only be performed by the superuser "Administrator". A database manager knowing the overall structure of the database well must be involved in analyzing data identified to violate data integrity.
- In case any data needs to be transferred to a project, this is considered to be a separate action. For example, if you want to recover a module into a project, this is like creating a module, covered by the feature "Formal modules". If that module contains data, you must review that data in order to ascertain the data are correct.

Similar remarks apply to the using the function "Comparing and restoring data from the module history": restoring means changing the data, and in this respect is like editing, covered by the feature "Editing formal modules".

6 Additional Considerations

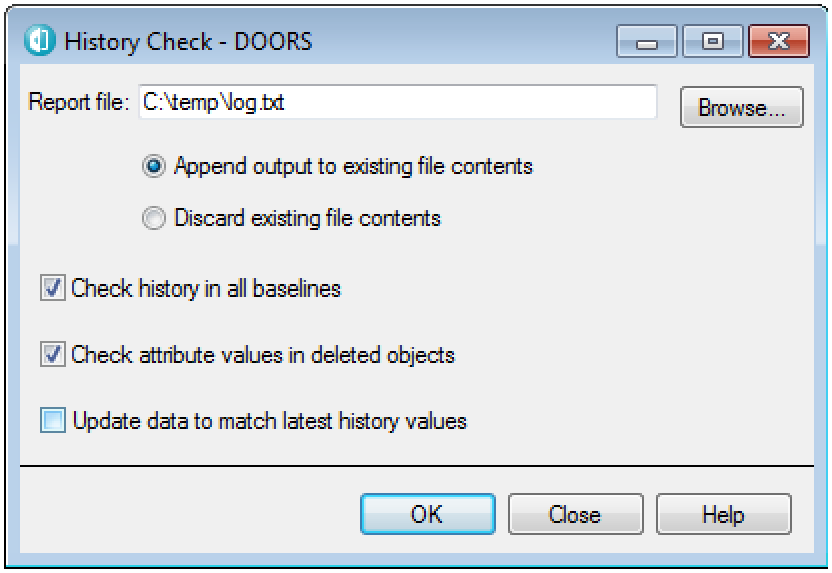
6.1 History check

It is recommended to compare the current version with the history using standard DOORS functionality as this feature is developed and tested and is covered by the present certificate.

DOORS provides the standard function "Comparing and restoring data from the module history" (cf. the corresponding section of **[DOORS User Manual]**) for this task. Execution of this function creates a report listing details of any discrepancies that are found. You can recover any lost data that is identified.

Please note that the function "Comparing and restoring data from the module history" can only be executed by the Administrator user, as this is the only user who has unconditionally unrestricted access to the history.

In the dialog of the history check select the options according to your preferences.



History Check - DOORS

Report file: C:\temp\log.txt

● Append output to existing file contents

○ Discard existing file contents

☑ Check history in all baselines

☑ Check attribute values in deleted objects

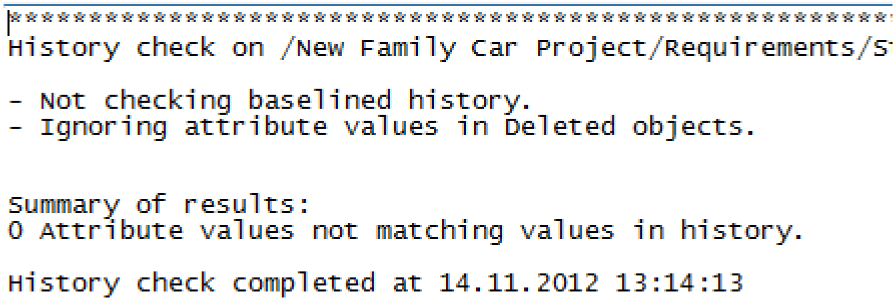
☐ Update data to match latest history values

Figure 14: Dialog for performing the DOORS history check

- The output is written to the file specified as "Report file"
- Select "Append output ..." if you want to keep previous reports
- Select "Discard ..." if you want to overwrite previous reports
- Select "Check history ..." if you want to compare against the history starting from the creation of the module. If you do not select this option, the comparison starts at the last baseline.
- Select "Check attribute values ..." if you want to include deleted objects in the comparison
- Select "Update data ..." if you want to change the current version to match the history values.

Make sure only to select the update option if you have reviewed before what changes you are about to make. By selecting the update option, you edit the module, and you must apply the corresponding checks prescribed by your development process.

The following figure shows a sample report generated by the history check.



```
********************************************************
History check on /New Family Car Project/Requirements/S

- Not checking baselined history.
- Ignoring attribute values in Deleted objects.


Summary of results:
0 Attribute values not matching values in history.

History check completed at 14.11.2012 13:14:13
```

Figure 15: Sample report of the DOORS history check

It is recommended to execute this function regularly. Execution may be automated in regular intervals, e.g. monthly or weekly, or it may be scheduled at certain milestones, as e.g. review or delivery.

6.2 Using redundant information

Providing the same pieces of information several times independently is a great way to prevent and detect errors. The same is true for calculating some figures in independent process steps several times. Agreement of the results is a strong indication that the information is correct, else the same error must occur in each calculation, or in every place where the information is provided. These are events with very low probability.

An example for providing the same piece of information twice is the following. When capturing review comments (in an attribute or in a discussion) users include the requirement ID in their comments. If data corruption occurs, or the user reviewed the wrong requirement, or the user logged comments on the wrong requirement, this mismatch will be detected in the review of the comments.

.

6.3 Perform a formal review

In safety critical context performing formal reviews is a must. Formal reviews are the most important means to ensure that inconsistent or incorrect information is detected, and the findings of reviews provide the basis for avoiding errors.

This manual does not aim at describing a complete target review process. Just some steps will be listed for consideration in the process, intending to increase the probability to detect tool related errors.

- Review comments must be captured in a safe way. This may include having separate attributes for each reviewer to comment on requirements. Reviewers should also capture the ID of the requirement in their comments. Thus errors in the comments as well as corruption can be identified easier.
- Review must include the review of traceability, i.e. affirm the validity of links. DOORS supports this step with traceability columns, cf section "Links and traceability > Adding a traceability column" of **[DOORS User Manual]**. Traceability columns display contents of linked objects along with contents of the current object, supporting a specialist review.
- Likewise, external links must be reviewed. For external links, the URL of the external item must be saved in an attribute.

It is worth noting that it is essential to detect errors prior to using information for further development. It cannot be expected that any error will be detected right after it has occurred as typically the detection means are too demanding to be applied permanently. So a formal review has the function to detect all errors that have occurred undetected.

6.3.1 Sample Requirements Review Checklist

The checklist below is provided as guidance. Each group should customize this to their own internal standards. Check list items that pertain specifically to ISO 26262 compliance have been noted. Not all check list items will pertain to requirements at all levels (e.g., System requirements will not trace to code) and users should either have customized check lists for requirement types or that ability to mark check list items “Not Applicable”.

In many cases users will want to represent their requirements review check lists (and other check lists) in electronic form to better manage the information.

Work Product(s) under Review			
Product / System name			
Product/Name ID			
Configuration ID			
Product type	**Requirements Data**		
Product(s) Owner			
Reviewer(s)			
Review date			
Standard(s) applied			
	Approved **Y/N**	**Item**	**Required actions / Comments**
Document Organization			
□		Does the requirements specification clearly specify it’s release date, authors, applicable project, relevant dates and revisions?	
□		Are terms, acronyms, and abbreviations defined?	

□		Are referenced documents and standards identified? Are references accurate and correct?	
□		Are all includes tables and figures referenced?	
□		Are functional and quality of services requirements organized by use case?	
□		Is each use case clearly characterized as to its scope and content?	
□		Are all use cases independent to a large degree?	
□		Is each functional or quality of service requirement clearly mapped to a use case?	
□		Is the requirements specification under configuration control?	
Overview			
□		Are externally visible operational modes identified and defined?	
□		Startup and initialiization	
□		Are requirements written so as to be clear, unambiguous, and testable?	
□		Is internal design detail avoided?	

□		Do the requirements provide a basis for thorough testing?	
□		Is internationalization addressed appropriately in the requirements?	
□		Are software partitioning requirements specifically addressed?	
□		Is there traceability from the requirements in this specification to ones above (e.g. system requirements or stakeholder requirements)?	
External Interfaces			
□		Are the actors with which the system interacts identified and their interfaces specified?	
□		Add all services within an interface defined in terms of the required data and control transformations engendered?	
□		Are external interfaces constrained with quality of service requirements including?	
□		- Worst case performance of response?	
□		- Worst case frequency of activation or arrival?	

□		- Error detection and correction?	
□		Is all data passed across external interfaces identified and specified logically?	
□		Are all important data values passed across interfaces specified in terms of value ranges, necessary fidelity and accuracy?	
□		- Value ranges	
□		- Accuracy	
□		- Fidelity	
□		- Unit of measure	
□		- Base type or logical structure	
□		- Physical representation	
□		- Source and/or destination	
Completeness			
□		Are all functional capabilities addressed within the requirements?	
□		Are all functional requirements defined in terms of the data and control transformation engendered?	
□		Are all relevant parametric requirements (e.g. heat, weight, size, power) addressed in the requirements?	
□		Are all functional requirements constrained by quality of service	

		requirements, such as	
□		- Timing	
□		- Performance	
□		- Frequency	
□		- Range and extent	
□		- Accuracy	
□		- round off and error handling	
□		- over and underflow processing	
□		Are all internal design decisions avoided within the requirements and left to design?	
□		Are statements which are not normative requirements clearly identified to be not requirements?	
□		Are statements which are normative requirements clearly identified as requirements?	
Correctness			
□		Is each requirement clear, unambiguous, concise, and testable? Per ISO 26262:2011-8 Clause 6.4.2.4	
□		Are the requirements free of conflict? Per ISO 26262:2011-8 Clause 6.4.2.4	
□		Do any requirements duplicate others. Per ISO 26262:2011-8 Clause 6.4.2.4	

□		Are all requirements within the scope of the specification?	
□		Are all requirements free from grammatical and syntactic errors?	
□		Are all identified safety concerns addressed by requirements?	
□		Are all identified reliability concerns addressed by the requirements?	
□		Are all security and information assurance concerns addressed by the requirements?	
□		Are all requirements likely to be implementable and achievable?	
□		Are all related requirements cross referenced and is that cross referencing correct?	
Quality of Service			
□		Are all functional requirements qualified with all appropriate quality of service requirements, such as	
□		- Frequency	
□		- Period	
□		- Jitter	
□		- Minimum inter-arrival time	
□		- Response performance	

□		- Accuracy	
□		- Fidelity	
□		- size complexity	
Traceability			
□		Is each requirement traceable to requirements that precede it (e.g. do software requirements trace to system requirements)? Per ISO 26262:2011-8 Clause 6.4.3.2	
□		Is each requirement traceable to one or more test cases that address it?	
□		Is each requirement traceable to one or more design elements that specify how it will be realized? Required for Safety Requirements per ISO 26262:2011-8 Clause 6.4.2.3 and 6.4.3.2	
□		Is each requirement traceable to lines of source code that implement it?	
Usability			
□		Are the user workflows involve the use of the required capabilities defined?	
□		Do the requirements specify the user interaction with the system in terms of	
□		- methods used for interaction	

□		- control layouts (if any)	
□		- display layouts (if any)	
□		- coupling of control usage	
□		- error and exception reporting	

6.4 Returning and checking imported data

A general problem when exchanging data is that you never can be sure that no data has been lost or modified during export or import. A good way to solve this problem is to double check. Double checking means employing redundant process steps.

An example of a simple check is to have the sender document the number of requirements being sent. The receiver will check that the number of requirements received is equal to the number sent. If the numbers are different, the receiver and sender will need to discover and correct the reason for the discrepancy.

Locks on the ReqIF data are not contained in the spreadsheet files, so the information exchanged is not identical in this respect. But errors in locks will be detected independently with high probability, cf. sections 5.5.1 and 5.5.2.

7 Specific Use Cases

In this section we present representative use cases, covering the main features which are relevant for most of the users. It is expected that these use cases will serve as examples or templates as each project will have to identify, document, and analyze its specific use cases.

Each subsection describes one use case, specifying the following information:

- title of the use case - is provided in the title of the subsection
- description - a general description of the use case is given in the introductory paragraph
- artifacts - both input artifacts and output artifacts are listed
- process description - listing the steps constituting the process.
- features used - based on the process description
- checks used - summarizes what checks are applied, giving rationale for the detection probability assigned to each feature
- TCL - determination of the tool confidence level for the use case, assuming all checks identified in the preceding subsection are employed

In a concluding step the TCL for the tool DOORS must be calculated from the TCLs of all use cases reflecting a costumer's process. According to the rules of the ISO 26262 the TCL for DOORS is the lowest TCL of all use cases. TCLs range from TCL1 (highest), to TCL2, to TCL3 (lowest). It is the responsibility of the user to validate that the generic TCL identified in Section 5 is valid for their specific usage of DOORS. If it is not, the TCL for that feature should be documented according to the project specific TCL. If the process defined in Section 4.5 of this document is followed, then all features in scope of this manual may be used on projects up to ASIL “D” or SIL”3”.

This manual does not aim at compiling a complete list of use cases as defining the relevant use cases for each project is the responsibility of the user. If needed, IBM Rational Services can give assistance in preparing the evaluation report. The use cases defined in the sequel illustrate how to map use cases to the list of features provided in the preceding chapter, and how to rate the use cases based on this mapping.

7.1 *Use Case: Derive requirements*

This use case specifies the derivation of requirements from input requirements.

Deriving requirements is typically done at several levels: system requirements are derived from stakeholder requirements, system requirements are allocated to subsystems, requirements are allocated to specific components (like hardware and software).

For each input requirement one or more output requirements are created. Additional information needed to support the process is captured in attributes of the requirement. In order to provide justification for the derived requirement links to the corresponding input requirements are created. The traceability thus created is represented in specific views on which traceability reports may be based.

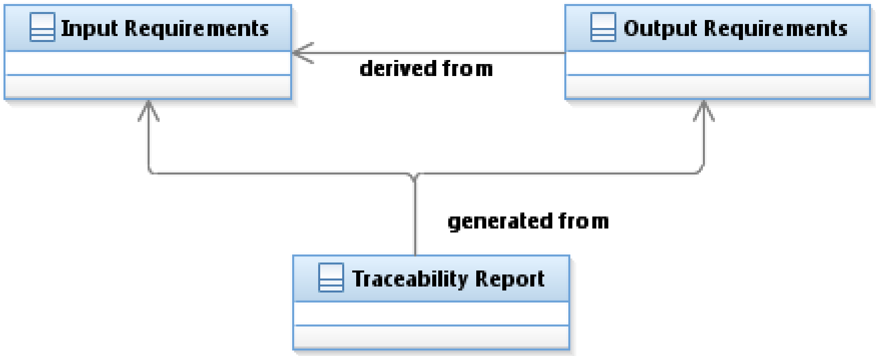


Figure 16: Use Case: Derive requirements

7.1.1 Artifacts

The use case "Derive requirements" reads and writes the following artifacts.

- input:
  - input requirement specification
- output:
  - output requirement specification
  - traceability report

7.1.2 Process description

For "Derive requirements" the following steps will be carried out as a sample process.

- Locate input module and output module in the DOORS database explorer.
- Open input module in read-only mode, open output module in edit mode.
- Select the view "Requirements and priority" in the input module, displaying IDs, requirement text and priority.
- Select the view "Derive requirements" in the output module, displaying IDs, requirement text, and status.
- For each input requirement, respond with one or more output requirements. Write the corresponding statement into the output module.
- For each requirement created, establish a link to the input requirement.
- For each requirement created, set the status to "Ready for review" when you finished working.
- After completing your work, switch to the view "Traceability" to display input requirements beside the output requirements and review its contents.
- After finishing the review, create a traceability report document using the Rational Publishing Engine. Check that the export is complete.

After completing a development phase, the process plans to do the following actions.

- A formal review on the contents and the traceability to the input requirements will be conducted.

7.1.3 Features used by "Derive requirements"

The use case "Derive requirements" thus uses the following features to support the process delineated in the last section.

Feature name	**Generic TCL from Section 5**
Access rights	TCL3
Controlling the data that is displayed in formal modules	TCL1
Database Explorer	TCL1
Editing formal modules	TCL1
Exporting data from formal modules using Rational Reporting for Document Generation	TCL3
Finding data in a DOORS database	TCL1
Formal modules	TCL3
Keyboard shortcuts	TCL1
Objects	TCL1
OLE objects and pictures	TCL1

Feature name	**Generic TCL from Section 5**
Standard links, external links, and collaboration links	TCL1

7.1.4 TCL for use case "Derive requirements"

In order to determine the tool confidence level for using DOORS for the use case "Derive requirements", the rules of ISO 26262 which are summarized in table 9 apply.

- Potential errors cannot be excluded when using DOORS, so the tool impact is TI2.
- The process defined in Section 4.5 is followed which includes an extensive peer review.
- Accordingly, for the sample process described above, the tool confidence level of DOORS for the use case "Derive requirements" employing the process sketched in section 7.1.2 is TCL3. However, as those features with a TCL3 have had an independent audit to verify the "Validation of the Software Tool" qualification option has been performed, they can be used in projects up to ASIL "D" or SIL "3".

7.2 Use Case: Create Test Specification

This use case describes the creation of a test specification based on the requirements managed in DOORS. Basically, customers may have test specifications in DOORS, in Microsoft Excel, IBM Rational Quality Manager, or in any other suitable tool.

In this use case, the test specifications are created and managed outside of DOORS. Depending on how customers managed their test specification, this use case must be adapted accordingly.

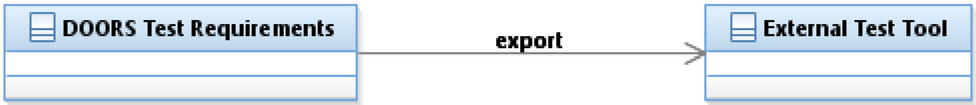


Figure 17: Use Case: Create Test Specification

Requirements are exported from DOORS and imported into the test tool by means of a tool connection that requires a DOORS client to run on the local machine. In DOORS a specific view for exporting is created. The test requirements are exported from this view to the test tool.

7.2.1 Artifacts

The use case "Create Module Test Specification" reads and writes the following artifacts.

- inputs:

    - model (system to be tested)
    - test plan
    - SW-requirements
- outputs:
    - test specification

7.2.2 Process description

For "Create Test Specification" the following steps will be carried out as a sample process.

- Locate the test requirements and the requirements module in the DOORS database explorer.
- Open the test requirements module and the requirements module in read-only mode.
- Select the view "Traceability to test requirement" in the requirements module. This view displays all requirements that are not traced to a test requirement. Check that there are no such requirements.
- Select the view "Traceability to requirements" in the test requirements module. This view displays all test requirements that are not traced to a requirement. Check that there are no such test requirements.
- Select the view "Test specification" in the test requirements module, displaying IDs, requirement text, verification method, "DOORS URL" (needed to establish a link back to DOORS), and "Test ID" (needed to identify the corresponding test item in the test tool), filtering requirements having "approved" as value for the attribute "Requirement Status".
- Export the view to an Excel file. Check that the export is complete.

After completing a development phase, the process plans to do the following actions.

- A formal review on the contents and the traceability to the input requirements will be conducted.

7.2.3 Features used by "Create Test Specification"

The use case "Derive requirements" thus uses the following features to support the process delineated in the last section.

Feature name	Generic TCL from Section 5
Access rights	TCL3
Controlling the data that is displayed in formal modules	TCL1
Database Explorer	TCL1
Exporting data from formal modules	TCL3
Finding data in a DOORS database	TCL1

Feature name	**Generic TCL from Section 5**
Formal modules	TCL1
Objects	TCL1
Rational DOORS URLs	TCL1
Standard links, external links, and collaboration links	TCL1

7.2.4 TCL for use case "Create Test Specification"

In order to determine the tool confidence level for using DOORS for the use case "Create Test Specification", the rules of ISO 26262 which are summarized in table 9 apply.

- Potential errors cannot be excluded when using DOORS, so the tool impact is TI2.
- The process defined in Section 4.5 is followed which includes an extensive peer review.
- Accordingly, for the sample process described above, the tool confidence level of DOORS for the use case "Derive requirements" employing the process sketched in section 7.1.2 is TCL3. However, as those features with a TCL3 have had an independent audit to verify the “Validation of the Software Tool” qualification option has been performed, they can be used in projects up to ASIL “D” or SIL “3”.

8 Mapping Reference Features to Safety Standards

This section indicates how the DOORS features described in section 5 map to both ISO 26262 and to IEC 61508.

8.1 Mapping to ISO 26262

ISO 26262 part 8, section 6 describes the specification and management of safety requirements as a supporting process to the standard process. It defines the scope of managing safety requirements to include managing requirements, obtaining agreement, obtaining commitment and maintaining traceability. The standard recommends the use of a suitable requirements management tool for managing safety requirements.

The requirements of ISO 26262-8.6 apply to several phases of the development process. It addresses the concept phase as well as the product development phase.

During the concept phase hazards are analyzed and risks assessed, which provide the basis for specifying safety goals. Details of this process are described in ISO 26262-3.7. In the sequel a functional safety concept is established, as described in ISO 26262-3.8.

During the product development phase, technical safety requirements are specified based on the functional safety concept. Details of this process step are described in ISO 26262-4.6. The technical safety requirements specification in turn is used to derive hardware safety requirements and software safety requirements, respectively, as described in ISO 26262-5.6 and ISO 26262-6.6.

The following table summarizes how the requirements of ISO 26262 part 8, section 6 are covered by features of DOORS.

- the column ***Clause*** refers to the corresponding section of ISO 26262, part 8. So "6.4.1" means ISO 26262 part 8, clause 6.4.1.
- the column ***Requirement*** expresses the requirement of that section
- the column ***DOORS feature*** lists the sections of this document and names the features covering the corresponding clause

Clause	Requirement	DOORS feature
6.4.1.1	Safety requirements shall be specified by natural language.	Natural language specification can be entered into the DOORS standard attribute "Object Text" and into any custom attribute based on the types "Text" or "String".
6.4.2.1	Safety requirements shall be unambiguously identifiable as safety requirements.	The definition of a suitable enumerated attribute allows the identification of safety goals and safety requirements. The attribute "Object Type" described in section 4.4.4.2 is an example how to do this.
6.4.2.2	Safety requirements shall inherit the ASIL from the safety requirements from which they are derived.	The attribute "ASIL" must be defined for all requirements modules, cf. section 4.4.4.1, and DOORS links must be established from any derived requirement to its sources. According to list item 2. of section 4.4.4.1 a DXL attribute assigns the highest ASIL of all sources.
6.4.2.3	Safety requirements shall be allocated to an item or an element.	Elements (ISO 26262-1, 1.32) or items (ISO 26262-1, 1.69) are defined to be systems, or parts of a system, or arrays of systems. The allocation demanded by this clause can be implemented in DOORS in several ways, e.g. by defining enumerated attributes listing elements and items, or by establishing links to a list of all elements and items.
6.4.2.5 a)	Safety requirements shall have a unique identification remaining unchanged throughout the safety lifecycle.	The attribute "Absolute Number", which is the numerical part of the "Object Identifier" of any DOORS object, is assigned to any DOORS object on its creation, and it can never be changed.
6.4.2.5 b)	Safety requirements shall have a status.	The definition of a suitable enumerated attribute tracks the status

Clause	Requirement	DOORS feature
		of requirements. The attribute "Requirements Status" described in section 4.4.4.5 is an example of how to do this.
6.4.2.5 c)	Safety requirements shall have an ASIL.	The definition of a suitable enumerated attribute allows ASILs to be assigned to safety goals. The attribute "ASIL" described in section 4.4.4.1 is an example of how to do this.
6.4.3.1 a)	The set of safety requirements shall have a hierarchical structure.	DOORS supports the setup of hierarchical structures consisting of projects, folders, and modules. Objects are arranged in a hierarchical structure in modules, representing the document structure. By implementing an information architecture reflecting the phases referred to by the ISO 26262, and by providing the proper document structure this requirement can be satisfied.
6.4.3.1 b)	The set of safety requirements shall have an organizational structure according to an appropriate grouping scheme.	In DOORS, folders are used to group modules. By implementing a suitable information architecture, e.g. based on the system architecture, this requirement can be satisfied.
6.4.3.1 c)	The set of safety requirements shall be complete.	Completeness of a set of requirements can only be ascertained by means of a review. Establishing DOORS links from any derived requirement to its sources and displaying information on linked objects in traceability columns supports any review activity.
6.4.3.1 d)	The set of safety requirements shall be externally consistent.	Like completeness, external consistency can only be ascertained by means of a review. Again, links and traceability columns can help to detect inconsistencies, as do filtering and searching.
6.4.3.1 e)	The set of safety requirements shall have no duplication of information within any level of the hierarchical structure.	Duplication of information can only be excluded by means of a review, too. Again, links and traceability columns as well as filtering and searching can are means employed to detect

Clause	Requirement	DOORS feature
		duplicates.
6.4.3.1 f)	The set of safety requirements shall be maintainable.	Provided a DOORS user has the appropriate access rights, they can modify or delete existing requirements in the current version of modules, as well as they can create new requirements. Baselined versions of modules are read-only for all DOORS users.
6.4.3.2 a)	Safety requirements shall be traceable with a reference being made to each source of a safety requirement at the upper hierarchical level.	This is satisfied by establishing links from the derived requirement to each of its sources. It is recommended to use a link module called "satisfies" for this type of link. DOORS links can be navigated in both directions, and they can be used to display any information of the sources at the derived requirement.
6.4.3.2 b)	Safety requirements shall be traceable with a reference being made to each derived safety requirement at a lower hierarchical level, or to its realization in the design.	This is also satisfied by establishing links from the derived requirements to each of its sources. The same type of link as above should be used here. Links now are used to display any information of the derived requirements at the source requirement.
6.4.3.2 c)	Safety requirements shall be traceable with a reference being made to the specification of verification.	Depending on the process and its support by tools this clause can be satisfied in a variety of ways. It is recommended to establish the specification of verification in separate DOORS modules and to create DOORS links from that specification to the corresponding requirements. It is essential to use a different link module for this type of relationship, which may be called "verifies". Collaboration links can be employed to establish traceability to a test management tool. Remember that integrations to test management tools that do not support collaboration links are not covered by the certificate.
6.4.3.3	An appropriate combination of the verification methods listed in table 2	This clause demands some kind of review to be performed. This activity

Clause	Requirement	DOORS feature
	shall be applied to verify that the safety requirements comply with the requirements in this clause and that they comply with the specific requirements on the verification of safety requirements within the respective parts of ISO 26262 where safety requirements are derived.	is supported by several DOORS features, including traceability columns, attributes, filtering, and baselines. The definition of the review process implies what features are needed for a specific situation
6.4.3.4	Safety requirements shall be placed under configuration management in accordance with Clause 7.	DOORS supports this clause by means of history, baselines, and baseline sets. Collaboration links to a configuration management tool may also be used. If the tool used for configuration management does not support collaboration links, an integration must be used instead. In that case the certificate does not apply.

8.2 Mapping to IEC 61508 Edition 2.0

The general framework of IEC 61508 (cf. IEC 61508-1, figure 1) describes the recommended process. According to this representation the clauses specifying the requirements that are relevant for requirements management are IEC 61508-1.5, IEC 61508-1.7.5, IEC 61508-1.7.6, and IEC 61508-1.7.10. The latter specifies the requirements for a requirements management tool like DOORS.

The following table summarizes how the requirements of those clauses are covered by features of DOORS.

- the column ***Clause*** refers to the corresponding section of IEC 61508, part 1. So "7.10.2.1" means IEC 61508 part 1, clause 7.10.2.1.
- the column ***Requirement*** expresses the requirement of that section
- the column ***DOORS feature*** lists the sections of this document and names the features covering the corresponding clause

Clause	Requirement	DOORS feature
7.10.2.1	The E/E/PE system safety requirements specification shall be derived from the allocation of safety requirements specified in 7.6 together with all relevant information related to the application. This information shall be made available to the E/E/PE safety-related system developer.	The different types of information will be captured in separate formal modules, each having the appropriate attributes defined to specify the information. Links will be created from requirements to the information they are derived from.

Clause	Requirement	DOORS feature
7.10.2.2	The E/E/PE system safety requirements specification shall contain requirements for the safety functions and their associated safety integrity levels.	Attributes like the "Object Type" and the "ASIL" described in sections 4.4.4.2 and 4.4.4.1, respectively, will be defined in order to classify requirements as safety requirements and to assign a safety integrity level to them.
7.10.2.3	The E/E/PE system safety requirements specification shall be made available to the developer of the E/E/PE safety-related system.	A DOORS user account must be created for each developer, and the appropriate access rights must be applied to the corresponding DOORS items.
7.10.2.4.c)	The E/E/PE system safety requirements specification shall be expressed and structured in such a way that it is expressed in natural or formal language and/or logic, sequence or cause and effect diagrams that define the necessary safety functions with each safety function being individually defined.	Requirements in DOORS are written in natural language in first place. Supporting information like diagrams may either be included as OLE objects, or else may be referred to using external links.
7.10.2.5	The specification of the E/E/PE system safety requirements shall contain the requirements for the E/E/PE system safety functions (see 7.10.2.6) and the requirements for E/E/PE system safety integrity (see 7.10.2.7).	Modules for each type of information can be established in DOORS, each providing the suitable set of attributes in order to specify the information.
7.10.2.6	The E/E/PE system safety functions requirements specification shall contain information characterizing the safety requirement (cf. IEC 61508-1, 7.10.2.6 a) - f)).	The information architecture of DOORS can be established accordingly, providing modules and attributes as required. A review must be performed in order to verify that each requirement is expressed appropriately. DOORS supports a review by defining attributes like "Review Status" and "Review Comments" as described in sections 4.4.4.6 and 4.4.4.7, respectively.
7.10.2.7.a)	The E/E/PE system safety integrity requirements specification shall contain the safety integrity level for each safety function and, when required, a specified value for the target failure measure.	An attribute like "ASIL" as described in section 4.4.4.1 can be defined, and a correspondingly defined attribute captures the target failure measure.

9 List of references

[DOORS Backup] cf. http://publib.boulder.ibm.com/infocenter/rsdp/v1r0m0/topic/com.ibm.help.download.doors.doc/pdf92/Backing_up_Rational_DOORS.pdf

[DOORS Communities] cf. http://www-947.ibm.com/support/entry/portal/forums_communities//software/rational/rational_doors

[DOORS Fix List] cf. http://www-01.ibm.com/support/docview.wss?rs=3601&uid=swg27024870&

[DOORS Information Center] cf. http://www-01.ibm.com/support/knowledgecenter/SSYQBZ/doors_family_welcome.html

[DOORS Reference Manual] synonym to **[DOORS User Manual]**, available at **[DOORS Information Center]**

[DOORS Support] cf. http://www.ibm.com/software/awdtools/doors/support

[DOORS System Requirements] cf. http://www-01.ibm.com/software/awdtools/doors/sysreqs

[DOORS User Manual] synonym to **[DOORS Reference Manual]**, available at **[DOORS Information Center]**

[DXL Reference Manual] available in the "Featured Documents" section of the DOORS Reference Manual, also cf. http://www-01.ibm.com/support/knowledgecenter/api/content/nl/en-us/SSYQBZ_9.6.0/com.ibm.doors.requirements.doc/topics/dxl_reference_manual.pdf and from the DOORS Help menu

[ECC] Error Correction Code memory, cf. http://en.wikipedia.org/wiki/ECC_memory and Error Detection and Correction, cf. http://en.wikipedia.org/wiki/Error_detection_and_correction

[HIS] Hersteller Initiative Software, cf. http://portal.automotive-his.de/index.php?lang=english

[IBM Fix Central] cf. http://www.ibm.com/eserver/support/fixes/

[IBM Product Information Center] cf. http://www.ibm.com/support/publications/us/library/

[IBM Rational Open APARs] cf. http://www-01.ibm.com/support/docview.wss?uid=swg21461170

[IBM Support Handbook] cf. ftp://ftp.software.ibm.com/software/server/handbook/webhndbk.pdf

[IBM Support Home] cf. http://www-947.ibm.com/support/entry/portal/overview

[IBM Support Subscriptions] cf. http://www-01.ibm.com/software/support/einfo.html

[ISO 26262] ISO26262 Road Vehicles –Functional safety–, of International Organization for Standardization, First edition of January 2011. It consists of ten parts.

[IEC 61508] IEC61508 Functional safety of electrical/electronic/programmable electronic safety-related systems, of International Electrotechnical Commision, Edition 2.0 of April 2010. It consists of seven parts.

[OMG] Object Management Group, cf. http://www.omg.org/

[RAID] Redundant Array of Independent Disks, cf. http://en.wikipedia.org/wiki/RAID

[VSS] Microsoft Volume Shadow Copy Service, cf. http://msdn.microsoft.com/en-us/library/windows/desktop/bb968832%28v=vs.85%29.aspx or http://en.wikipedia.org/wiki/Shadow_Copy

10 List of Figures

11 Tool Error List by Feature

This section lists summaries of generic potential errors for each feature of DOORS that the certificate applies to.   These generic potential errors are a result of a Tool Classification project done by IBM Rational.

In this section, the features of DOORS are organized according to the following topics:

- Features at database level
- Features for documents
- Features for displaying information
- Features for links
- Features for data exchange
- Additional features
- Features for customizing DOORS

Each subsection provides a description of the corresponding **feature** and lists the actions that are supported by the feature. If appropriate, the feature is differentiated from related features. For each feature a table summarizes the potential **errors** (derived from an error analysis activity similar to hazard analysis) related to that feature.

Listing errors for each feature introduces a certain degree of redundancy (as the same error can exist for more than one feature and one check may cover several errors) but it makes it easier for the reader to consume as all information relating to a single feature is mentioned in a single section.

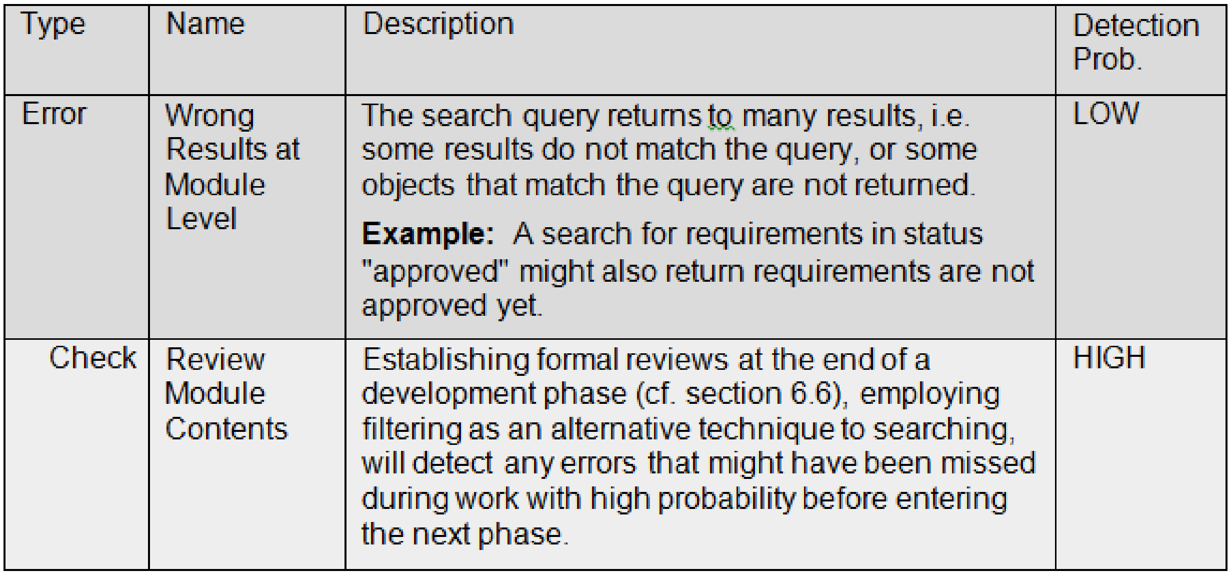
This analysis is done in a generic sense, apart from a project's or company's actual usage of DOORS. According to the ISO 26262 standard, each company or project is required to analyze the specific features of DOORS in the context of how they use it. Accordingly, this information serves as a basis for that activity and will hopefully make it easier, but it does not replace it.

The errors below focus on potential tool related errors and does not mean that these errors exist or have been found. As with any error/hazard analysis, the intent was to identify what errors might occur. For example, a hazard analysis of an automobile may identify the potential error or hazard of the brakes failing. This does not mean that the breaks actually do or will fail in the automobile, but the potential for that event is identified and mitigated against in the design process.

Each error is assigned a detection probability, which by default is considered LOW (corresponding to TD3). If MEDIUM (corresponding to TD2) or HIGH (corresponding to TD1) detection probability is claimed, a **rationale** must be provided. This information is given in the tables as well.

Checks may be used to increase the detection probability of an error. Each user should decide what, if any, checks should be applied in their environment and process to increase the probability of detection or decrease probability of occurrence of each potential error.

An example of such a check is adding to a process check of “Review Module Contents” to increase the probability of detection of the “Wrong results at the module level” error.



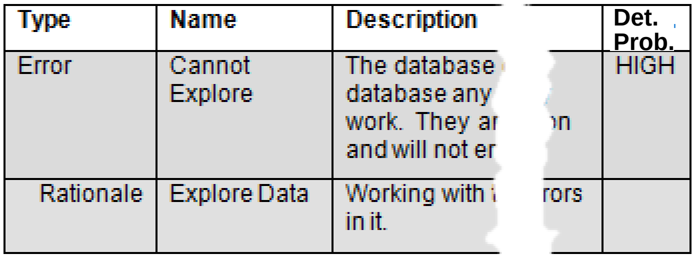
Type	Name	Description	Detection Prob.
Error	Wrong Results at Module Level	The search query returns to many results, i.e. some results do not match the query, or some objects that match the query are not returned. **Example:** A search for requirements in status "approved" might also return requirements are not approved yet.	LOW
Check	Review Module Contents	Establishing formal reviews at the end of a development phase (cf. section 6.6), employing filtering as an alternative technique to searching, will detect any errors that might have been missed during work with high probability before entering the next phase.	HIGH

Figure 18: Example of using checks to increase probability of detecting errors

The errors that may occur for each feature are summarized in a table. Each table consists of the following columns.

- **Type:** specifies if the row describes an error or a rationale
- **Name:** short name of the error or rationale
- **Description:** a short description of the error or rationale
- **Prob.:** probability that the error is detected without employing checks. As a rationale explains the detection probability of the preceding error, no probability is listed for rationales.

The figure below illustrates how to read the tables.



Type	Name	Description	Det. Prob.
Error	Cannot Explore	The database database any work. They ar on and will not er	HIGH
Rationale	Explore Data	Working with i rors in it.	

Figure 19: Example: justification for HIGH detection probability

11.1 Features at database level

This section describes the features used to navigate the database to locate projects, folders, and modules.

11.1.1 Database Explorer

The database explorer allows DOORS users to navigate through the projects, folders, and modules in the database.

The feature "Database Explorer" supports the following actions.

- Cutting, copying, and pasting in the database explorer
- Deleting, undeleting, and purging
- Copying the structure of an existing module
- Entering paths
- Using favorites

For the feature "Database Explorer" the following potential errors were identified:

Type	Name	Description	Detection Prob.
Error	Cannot Explore	The database explorer does not explore the database anymore and the user cannot do any work. They are aware that there is a malfunction and will not enter new data. The user must assume all data since the last successful "Save" is lost or suspect.	HIGH
Rationale	Explore Data	Working with the database explorer shows trivial errors, like no contents at all, or closing the explorer window.	
Error	Wrong Exploration	The database explorer shows wrong results, e.g. some folders or modules might be missing.	HIGH
Rationale	Use During Work	The data is displayed graphically and verified during creation and further processing. Thus errors would be detected with a high probability, even if this is a manual step.	

11.1.2 Finding data in a DOORS database

Searches can either be initiated from the DOORS database explorer, or from within individual modules. As a result of a search from the database explorer a list of matching projects, folders or modules is displayed. Searching within a module sets the focus to the next matching object.

Within modules, usually filters are used instead of searching.

The feature "Finding data in a DOORS database" supports the following actions.

- Searching the database
- Searching and replacing text in modules
- Regular expressions for searching and filtering modules

- Going to a particular object in a formal module

Note that using filters is not covered by the present feature, but is described by the feature "Using Views" in section 5.3.2.

For the feature "Finding data in a DOORS database" potential errors were identified.

Type	**Name**	**Description**	**Detection Prob.**
Error	Too Many Results at Database Level	The search query returns too many results, i.e. some results do not match the query. **Example:** A search for items containing "system" in the name also returns the module "Acceptance test".	HIGH
Rationale	Work with Result at Database Level	The result of the query is used for further processing. Folders or modules will be opened, items that are not matching the query will be detected with high probability by users who are familiar with the data.	
Error	Result Missing at Database Level	Some projects, folders, or modules matching the query are not returned.	HIGH
Rationale	Use Search Results at Database Level	The result of the query is used for further processing. The absence of folders or modules will be detected with high probability by users who are familiar with the data.	
Error	Wrong Results at Module Level	The search query returns to many results, i.e. some results do not match the query, or some objects that match the query are not returned. **Example:** A search for requirements in status "approved" might also return requirements are not approved yet.	LOW
Error	Wrong Edit	The editor does not change the values as intended. **Comment:** this error applies to searching and replacing. Replacing is like editing, so this error must be included here	LOW

11.1.3 Projects and folders

Projects and folders are used to organize and structure the data in the database. They can be created anywhere in the database hierarchy. Compared to folders, projects have additional functions used for administration at database level.

The feature "Projects and folders" support the following actions.

- Access rights for projects and folders
- Locks
- Creating projects
- Creating folders
- Editing project and folder properties
- Converting a project or a folder

For the feature "Projects and folders" the following “potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Persistency Error	Projects or folders are not created as expected, or their structure is corrupted or changed. **Comment:** User errors of creating the wrong folder or using the wrong name are not considered here.	LOW

11.2 Features for documents

Documents are called "formal modules" in DOORS. Formal modules are composed of Objects, which may contain embedded objects, and they define the scope for attribute definitions. Changes to module contents are recorded in the module history.

11.2.1 Formal modules

Formal modules contain information in objects that are defined by their attributes.

The feature "Formal modules" supports the following actions.

- Creating modules
- Copying the structure of an existing module
- Managing access rights for formal modules
- Managing open modules
- Showing module statistics
- Selecting toolbars to show in modules

Editing contents of formal modules is not covered by this feature, but is described by the feature "Editing formal modules". The access to formal modules is controlled by the access rights it is assigned. Access rights are not covered by this feature, but are described by the feature "Access Rights".

For the feature "Formal modules" the following " potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Persistency Defect	Formal modules are not created as expected, or their structure is corrupted or changed.	LOW

11.2.2 Editing formal modules

DOORS formal modules consist of a hierarchy of objects, which in turn contain information, as e.g. headings, requirements, and background information.

Unlike the feature "Objects" this feature does not address single objects, but describes how the module supports managing a number of objects.

The feature "Editing formal modules" supports the following actions.

- Selecting an edit mode
- Creating editable sections in modules
- Locking and unlocking editable sections
- Using the object properties sheet to edit objects and attributes
- Editing in the module window
- Using forms to edit objects and attributes
- Creating and editing tables
- Inserting modules as tables
- Inserting symbols
- Undoing edits
- Checking spelling

For the feature "Editing formal modules" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Wrong Edit	The editor does not change the values as intended.	HIGH
Rationale	User Validation	The users validate the edited values during edit. Any error will be detected with high probability as they see how the data are changed.	
Error	Persistency Problem	The editor does not save the edited values as displayed.	LOW

11.2.3 Objects

An object is a database record. Objects are members of a formal module. They store their information in attributes that are defined within the module.

The feature "Objects" supports the following actions.

- Creating objects
- Selecting objects
- Copying objects
- Dragging objects
- How copy and move affect links
- Manipulating objects
- Deleting, undeleting, and purging objects

For the feature "Objects" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Persistency Error	Objects appear to be deleted on the client, but actually they are not deleted in the database.	LOW
Error	Wrong Object Hierarchy	Objects are created at or moved to the wrong location in the object hierarchy in a DOORS module.	LOW

11.2.4 Attributes

In DOORS, information about modules and objects is stored in attributes. This feature deals with the administration of attributes. Management of attribute values is addressed by the features "Editing formal modules" and "Objects".

The feature "Attributes" supports the following actions.

- Managing attribute types
  - Creating attribute types
  - Editing attribute types
  - Deleting attribute types
- Managing attribute definitions
  - System attributes
  - Locales for textual attributes
  - Creating an attribute definition
  - DXL attributes and layout DXL columns
  - Editing attribute definitions
  - Deleting attribute definitions

- Reusing attribute types and attribute definitions
- Copying attribute values
- Measuring the frequency of attribute values

For the feature "Attributes" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Corrupted Attribute	The definition of an attribute is corrupted in some way. The attribute type may be defined wrongly, the attribute may be defined in an unintended way, e.g. the scope or the features are not correct, or the attribute does not exist although it is expected to exist.	LOW

11.2.5 OLE objects and pictures

DOORS objects may contain OLE objects. OLE objects represent data managed by a separate application. DOORS may display the contents of that information, or it may display an icon.

DOORS can also represent OLE objects as pictures objects, which unlike OLE objects cannot be edited.

The feature "OLE objects and pictures" supports the following actions.

- Editing OLE objects
- Inserting OLE objects
- Inserting pictures
- Editing OLE object properties
- Recording history for OLE objects
- Handling unregistered OLE objects

As OLE objects are not modified by DOORS but by the separate application (e.g. Excel tables are edited with Microsoft Excel, pictures are edited with some graphics program), editing OLE objects is out of scope. This feature covers the handling of OLE data by DOORS, e.g. inserting and maintaining the link information.

For the feature "OLE objects and pictures" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Wrong OLE Created	The wrong OLE is inserted as OLE object or as picture object.	LOW

11.2.6 Tracking changes to DOORS data

DOORS records changes to objects and modules in a history. Based on history records and links users are supported to track changes to data, and to manage the impact that changes have to other items in the database.

The feature "Tracking changes to DOORS data" supports the following actions.

- Change bars
- Module history
- Redline markup in DOORS
- Suspect links and changed objects

For the feature "Tracking changes to DOORS data" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Wrong Tracking	A change in the data is not tracked or tracked wrongly, or an unchanged item is marked as changed.	LOW

11.2.7 Discussions

Discussions allow developers to exchange views about the content of a module or the content of an object in a module without needing modify access to the module or object. Discussions are presented as part of the properties of the object or module.

Please note that if you archive and restore a module, discussions about the module or objects in the module are not restored. Therefore it is strongly recommended to use discussions for informal exchange of information only, but not for formal reviews. If this recommendation is followed, this feature in general will not be safety critical, supporting the communication on information captured in DOORS objects or modules. Therefore the error model is not very detailed, subsuming all errors and checks in a single item each.

The feature "Discussions" supports the following actions.

- Creating discussions
- Controlling access for discussions in a module
- Viewing discussions and adding comments
- Closing, reopening, and deleting discussions

For the feature "Discussions" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Corrupted Discussion	The discussion has been corrupted, such that the content is lost, not readable or modified.	HIGH

Type	Name	Description	Detection Prob.
Rational	Working with Discussion	During the work with the discussions the errors will be detected. **Comment:** Since the discussions are based on a human understanding of the discussed items, tool errors like wrong content will be noticed with a high probability.	

11.3 Features for displaying information

DOORS holds the contents of modules in a database. Views are used as a means to control the way the data is presented to the users. Filters and sorting criteria may be saved as part of a view.

11.3.1 Controlling the data that is displayed in formal modules

DOORS allows users to define a layout for module data that is most suitable for a certain task. The layout includes columns, filters, sorts, and what levels of the hierarchy to display. A layout intended to be used again is saved as a view that can be accessed every time the module is open.

By default after opening a formal module the Standard view is displayed, showing the module explorer, the ID column, and the main column.

Users may save additional views containing e.g. additional columns, applied filters or sorts, changed size of the module window. Users may select views from a list. Each view may be set as default, so users are presented this view instead of the Standard view.

Filters let the user control the data that is displayed in a module by including or excluding objects according to simple or combined conditions. A filter may or may not be saved as part of a view.

The feature "Controlling the data that is displayed in formal modules" supports the following actions.

- Defining filters
- Regular expressions for searching and filtering modules
- Defining sorts
- Selecting the level of objects to show in modules
- Selecting the attribute to show in tables
- Inserting, editing, and removing columns
- Inserting graph columns
- Inserting icon columns
- Saving views
- Viewing, editing, and deleting views

For the feature "Controlling the data that is displayed in formal modules" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Wrong Layout	The view does not have the intended layout. **Comment:** This error comprises showing too many or too little columns, not applying the intended sorting criteria, and errors in layout DXL.	HIGH
Rationale	Visual Control During Work	The graphical data is verified during creation or further processing. Differences to the expected layout will be detected with high probability, even if this is a manual step.	
Error	Wrong data displayed	The view shows too many or too little objects.	LOW

11.4 Features for links

Traceability in DOORS is created using links. DOORS offers several features supporting the creation and the analysis of traceability. DOORS standard links establish bidirectional traceability within a single DOORS database. Link modules and linksets help setting up a schema forcing DOORS users to create links as intended. External links and DOORS URLs support navigation from and to DOORS, while collaboration links establish a relationship between DOORS information and information held in other Rational or non-Rational tools that have been set up correspondingly, such as e.g. design artifacts, test artifacts, or change management artifacts.

11.4.1 Standard links, external links, and collaboration links

Links are used to associate several pieces of information. DOORS supports several types of links depending on the nature of information.

- Standard links associate requirements in the same DOORS database with one another.
- External links associate requirements in a Rational DOORS database with an entity or resource that is outside the current DOORS database.
- Collaboration links associate requirements with artifacts in a server that has been set up using the Rational solution for Collaborative Lifecycle Management (CLM).

While external links provide a means of navigation from a DOORS database to resources outside of that database, the feature "DOORS URLs" allow users to navigate from resources outside of DOORS to a specific item in the DOORS database.

As external links and collaboration links leave the responsibility of DOORS, their usage is not covered by this certificate. If users want to make use of external links or collaboration links, it is their responsibility to ensure qualified usage of the application these links refer to.

If external links are used, it is the responsibility of the user to ensure that the items pointed to by external links exist and are maintained. If the external links are pointing to some other application, like Microsoft Work documents, it is also the responsibility of the user to make sure that this usage is covered by their process. If a user navigates to an external URL that is not available or not valid, then they will get an error message stating that the browser cannot display the webpage, or it cannot find the file. The user then needs to determine if the link was set correctly (if not previously validated and baselined), or if the target information was moved incorrectly and has to be restored, or if the broken link has to be fixed by other means.

The feature "Standard links, external links, and collaboration links" supports the following actions.

- Creating links
- Creating multiple links
- Creating external links
- Creating collaboration links
- Creating links by attribute
- Editing links
- Deleting links from multiple objects

This feature describes how to maintain individual links. Configuration of a link schema and analyzing traceability are not covered by this feature, but by the features "Link modules, linksets, and linkset pairings" and "Links and traceability", respectively.

For the feature "Standard links, external links, and collaboration links" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Broken Link	The link points to an invalid target.	LOW
Error	Incorrect Link	A link points to a wrong item, or a link has been lost.	LOW

There is a "Stale link detection" utility from the custom DOORS menu (cf. section 6.3) that can be used to increase the probability of detecting errors with links. Also, working with links during reviews or other work will also increase the detection probability of this error.

11.4.2 Link modules, linksets, and linkset pairings

For standard links, DOORS supports setting up and enforcing a schema for linking.

Link modules store information about standard Rational DOORS links. For each type of link, e.g. "satisfies" linking requirements and "tests" linking test cases to requirements, a different link module is used. Within each link module, the information is subdivided into linksets, each representing relationships from one particular module to another.

Each formal module can be configured to specify what target modules and what type of link may be used for establishing links.

The feature "Standard links, external links, and collaboration links" deals with using links, whereas the present feature "Link modules, linksets, and linkset pairings" is concerned with administrative aspects of standard DOORS links.

The feature "Link modules, linksets, and linkset pairings" supports the following actions.

- Creating link modules
- Defining default link modules
- Editing link modules
- Viewing, creating, and deleting linksets
- Creating link attributes
- Creating linkset pairings
- Managing access rights for link modules

This feature describes the configuration of a link schema. Maintaining or analyzing traceability is not covered by this feature, but is described by the features "Standard links, external links, and collaboration links" and "Links and traceability", respectively.

For the feature "Link modules, linksets, and linkset pairings" the following potential errors were identified.

Type	**Name**	**Description**	**Detection Prob.**
Error	Wrong Link Configuration	Link modules or linksets are missing, linkset pairings are not configured, or link attributes are defined inappropriately.	LOW
Error	Wrong Access for Link Modules	Access rights for link modules are set incorrectly. **Comment:** As the configuration of links is crucial it is worth applying a restrictive access rights schema, cf. section 4.5.2. This also is the reason to consider access rights for link modules separate from access rights for other DOORS items.	LOW

11.4.3 Links and traceability

Links between objects provide the basis for several types of analysis, supporting users in assessing requirements coverage and completeness of the specification. Traceability also helps in managing change by quickly identifying what impact a proposed change has on the project.

The feature "Standard links, external links, and collaboration links" deals with using links, whereas the present feature "Links and traceability" is concerned with navigating and analyzing links.

The feature "Links and traceability" supports the following actions.

- Analyzing links
- Adding a traceability column
- Analyzing links using the traceability explorer

This feature describes the analysis of traceability. Maintaining traceability and configuration of a link schema are not covered by this feature, but they are described by the features "Standard links, external links, and collaboration links" and "Link modules, linksets, and linkset pairings", respectively.

For the feature "Links and traceability" the following potential errors were identified.

Type	**Name**	**Description**	**Detection Prob.**
Error	Incorrect Link	A link points to a wrong item, or a link has been lost.	LOW

A formal review can be used to determine that traceability is correct and complete prior to baselining to increase the probability of detecting this error.

11.4.4 DOORS URLs

Each item of the DOORS database, including the database, projects, folders, modules, baselines, and objects has a unique identifier, which is expressed as a URL. The URL of each item is stored in its properties sheet, and can be copied from there. The DOORS URL is a means to navigate from other applications, as e.g. a Microsoft Word document, to the specified DOORS item.

If DOORS URLs are used to navigate from some other application to DOORS items, it is the responsibility of the user to make sure that this usage is covered by their process.

The feature "DOORS URLs" supports the following action.

- Following DOORS URLs

For the feature "DOORS URLs" the following potential errors were identified.

Type	**Name**	**Description**	**Detection Prob.**
Error	No Correct URL	The URL is no URL, i.e. it does not point to a valid object, or it points to a wrong item. **Comment:** corruption of URLs which possibly occurs outside of DOORS are not covered by this error. The user must account for this error in the definition of the process for the other tool.	LOW

After creating a URL, a process check to ‘execute” that URL can be added to increase the detection probability of this error.

11.5 *Features for data exchange*

There are several ways to exchange data between DOORS and other tools, or to extract data from DOORS. Each of these features will be described in the subsequent sections.

11.5.1 ReqIF

The Requirements Interchange Format (ReqIF, also called RIF up to 2010) is used to exchange requirement information between DOORS databases.

DOORS data can be send for editing to another DOORS database. After the data has been edited, it is returned to the originating DOORS database, and, if appropriate, merged with the original data. Returning the data and merging it is a two-step process.

Please note that the DOORS menu items as well as the DOORS reference manual use "RIF" for Requirements Interchange Format for the versions analyzed.

The feature "ReqIF" supports the following actions.

- Sending and receiving RIF files
- RIF locks on local data
- Creating RIF definitions
- Editing RIF definitions
- Deleting RIF definitions
- Exporting RIF packages
- Importing RIF packages
- Merging RIF packages
- Recovering RIF locks

For the feature "ReqIF" the following potential errors were identified.

Type	**Name**	**Description**	**Detection Prob.**
Error	Corrupted RIF Data	The RIF data package has lost some contents, or its content has been modified during export or import, e.g. by changing the contents of some attribute values.	LOW
Error	Locked RIF Data	The export or import has wrong lock information for the data.	HIGH
Rationale	Work With RIF Data	If the ReqIF package has inaccurate lock information, either the sending or the receiving database cannot work as intended because of missing write access. This error will be detected with high probability during work.	

Using a tool such as PKZIP which includes a checksum on data prior to sending a RIF package can increase the probability of detecting any corruption of data during exchange of RIF data.

11.5.2 Data partitions

Data partitions are a means to exchange data with users of a different DOORS database. In the away database the data partitioned out may either be viewed or edited and then returned to the home database. Data that may be edited by the users in the away database is locked in the home database until it is returned and rejoined, or recovered.

The feature "Data partitions" supports the following actions.

- Sending and receiving partition files
- Creating partition definitions
- Editing partition definitions
- Deleting partition definitions
- Exporting partition definitions
- Importing a partition
- Showing exported and imported partitions
- Adding data to imported partitions
- Creating a sync file from an imported partition
- Synchronizing an exported partition
- Returning an imported partition
- Rejoining a partition
- Recovering a partition

For the feature "Data partitions" the following potential errors were identified.

Type	**Name**	**Description**	**Detection Prob.**
Error	Corrupted Partition	The partition file has lost some contents, or its contents has been modified during export or import, e.g. by changing the contents of some attribute values.	LOW
Error	Locked Partition	The export or import has wrong lock information for the data.	HIGH
Rationale	Work With Partition	If the partition has inaccurate lock information, either the sending or the receiving database cannot work as intended because of missing write access. This error will be detected with high probability during work.	

11.5.3 Exporting data from formal modules

DOORS exporters extract data from formal modules to other applications or formats. The contents of the export are generally controlled by selecting a view for export.

The feature "Exporting data from formal modules" supports the following actions.

- Exporting to Microsoft Word
- Exporting to Microsoft Excel
- Exporting to Microsoft Outlook
- Exporting to PowerPoint
- Exporting to HTML
- Exporting to plain text
- Exporting to RTF
- Exporting to spreadsheet
- Exporting to FrameMaker

For the feature "Exporting data from formal modules" the following potential errors were identified.

Type	**Name**	**Description**	**Detection Prob.**
Error	Incomplete Export	The export does not contain all intended information. **Comment:** this includes empty output.	LOW
Error	Wrong Export	The content of the export is not as intended.	LOW
Error	Corrupted Export	The content of the export is corrupted or incomplete, so that it cannot be opened by the other tool.	HIGH
Rationale	Open Export With Tool	The exported document is opened with a tool. Since there is a tool that uses the document syntactic errors will be detected with high probability.	

Comparing the count of items exported from DOORS with the number of items in the exported tool can help increase the probability of an incomplete or wrong export.

11.5.4 Exporting data from formal modules using Rational Reporting for Document Generation

Rational Reporting for Document Generation extracts data from DOORS modules into templates, defining the layout of the report. The report may be generated in Microsoft Word format, in PDF format or in HTML format.

The feature "Exporting data from formal modules using Rational Reporting for Document Generation" supports the following actions.

- Configuring the Document Generation defaults for the module
- Exporting to Microsoft Word using Rational Reporting for Document Generation

- Exporting to PDF using Rational Reporting for Document Generation
- Exporting to HTML using Rational Reporting for Document Generation

For the feature "Exporting data from formal modules using Rational Reporting for Document Generation" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Incomplete Export	The export does not contain all intended information. **Comment:** this includes empty output.	LOW
Error	Wrong Export	The content of the export is not as intended.	LOW
Error	Corrupted Export	The content of the export is corrupted or incomplete.	HIGH
Rationale	Open Export With Tool	The exported document is opened with a tool. Since there is a tool that uses the document syntactic errors will be detected with high probability.	

Comparing the count of items exported from DOORS with the number of items in the exported tool can help increase the probability of an incomplete or wrong export.

11.5.5 Printing and reports

The contents of formal modules can be sent to a printer. DOORS uses page setups to control the layout of the printout. DOORS always prints views, i.e. those objects and those columns that are displayed. A report combines a page setup and a view, thus providing a means to print from the database explorer without opening the formal module first.

The feature "Printing and reports" supports the following actions.

- Creating, editing, and applying page setups
- Printing modules
- Creating and managing reports for printing

For the feature "Printing and reports" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Wrong Content	The contents of the printout is wrong.	LOW

Reviewing the report can increase the detection probability of this error.

11.5.6 Importing data into formal modules

DOORS importers populate formal modules with data from other applications.

The feature "Importing data into formal modules" supports the following actions.

- Exporting from Microsoft Word to DOORS
- Editing imported style information from Word
- Importing plain text files
- Importing rich text format files
- Importing spreadsheets
- Importing FrameMaker files

For the feature "Importing data into formal modules" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Wrong Import	The data is not imported, or imported incompletely or wrongly.	LOW

11.6 Additional features

DOORS provides several additional features which do not group to any of the preceding topics.

11.6.1 Authenticate

DOORS provides specific access to the data to users. So each user has to authenticate providing credentials before achieving access to DOORS data. Credentials may be supplied directly by entering user name and password, or they may be read from a smart card like CAC (Common Access Card).

Only registered users can authenticate at a DOORS database. Every user who has authenticated successfully is subject to the DOORS access rights which control what they are allowed to do in the DOORS database.

For the feature "Authenticate" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Authentication Failure	A registered user cannot authenticate. **Comment:** this is annoying, but not a safety issue.	HIGH
Rationale	Attempt to Work	The user will not be able to work, thus the error has high detection probability.	

11.6.2 Access rights

Access rights control what users can do to each item in a DOORS database. Access rights can be assigned to individual users or to groups of users.

The feature "Access rights" supports the following actions.

- DOORS groups
- Access rights and inheritance
- Propagating additional access rights with create access

For the feature "Access rights" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Wrong Access Rights	The user has wrong access rights and can especially do too much.	LOW

11.6.3 Baselines

In DOORS, a baseline is a read-only version of a module. The baseline captures the status of the module at a certain time and preserves it until the module is deleted.

Baselines include the module history:

- Information about all the attribute definitions and types that have been created, deleted, or edited since the most recent baseline.
- Information about all the objects that have been created, deleted, or edited since the most recent baseline.
- Information about every module session (every time the module has been opened) since it was first created.

It is recommended to create baselines regularly to improve performance. The history file is removed from the module, and a file that contains the baseline, together with all the history information is created. So the history information remains in the system, it can be accessed from the baseline.

Several modules that form a unit for project planning and project management purposes can be combined in a baseline set definition. Baseline sets can be created for each phase of the project from this baseline set definition. A baseline set presents a snapshot of the data in the modules and the links that existed between them when the baseline set was closed.

The feature "Baselines" supports the following actions.

- Creating baselines at project level (baseline sets)
    - Creating baseline set definitions
    - Copying baseline set definitions
    - Removing baseline set definitions
    - Creating baseline sets
    - Closing baseline sets
- Creating baselines
- Copying baselines

- Opening baselines
- Comparing baselines
- Deleting baselines
- Electronic signature

For the feature "Baselines" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Wrong Baseline	The baseline is incomplete or contains wrong or corrupted data.	LOW
Error	Link to Wrong Version	Two modules are already included in a baseline set, but links do not go between the baselined versions of the module, but from one baseline version to the current version. **Comment:** this error refers to the situation when links are expected to go between baselined versions. This is ***not*** the intermediate situation when one module is already in the baseline set while the other one is not	LOW

11.6.4 Keyboard shortcuts

Keyboard shortcuts allow DOORS to be operated avoiding the mouse.

The feature "Keyboard shortcuts" supports the following actions.

- Using control keys
- Applying rich text formatting
- Using function keys
- Using keypad keys
- Using navigation keys

For the feature "Keyboard shortcuts" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	Unintended Action	The action of the shortcut is not the intended one, either no action or a wrong one is executed.	HIGH
Rationale	User Validation	During the work with the editor the user will see the effects of the shortcuts and would detect deviations with a high confidence.	

11.7 Features for customizing DOORS

DOORS allows users to extend its functionality by providing the scripting language DXL, the DOORS Extension Language. There are two options to account for using DXL.

- DXL programs are considered as separate tools, for which the author can follow a procedure like the one described for DOORS in this report. In that case, the analysis is performed for a tool chain, consisting of DOORS and the DXL tool.
- The strict procedure is not followed for the DXL program. In that case errors are considered to be potential errors of DOORS, and checks can be used to detect these errors.

This covers the core DXL as provided by DOORS. Programs written using the DXL language are outside the scope of this safety manual.

11.7.1 Using DXL (the DOORS Extension Language)

The DOORS eXtension Language (DXL) is a scripting language having a syntax like C and C++. It can be used to control and extend DOORS functions. Typical uses of DXL include

- Automate routine or complex tasks, such as calculating attribute values.
- Respond to events by triggering custom programs.
- Add your own options to DOORS menus.

A full description of DXL is given in the **[DXL Reference Manual]**, which is available at the "Featured Documents" section of the reference manual as a PDF and from the DOORS Help menu.

The feature "Using DXL" supports the following actions.

- DXL encryption
- Setting up DXL security
- The DXL library
- Converting layout DXL to attribute DXL

Any DXL code written by the user or a third party is not covered by the certificate. It is the responsibility of the user to make sure that any DXL add-on used in safety critical context meets the requirements of the standards, including analyzing use case coverage. DXL sample programs delivered as means to illustrate the usage of DXL are explicitly included in the preceding statement.

For the feature "Using DXL (the DOORS Extension Language)" the following potential errors were identified.

Type	Name	Description	Detection Prob.
Error	DXL Program Error	Any Error that causes sample DXL scripts to fail such that the results are wrong.	LOW
Error	Wrong DXL	The wrong DXL is executed as the menu item does not address the intended files.	LOW

Each user needs to verify their use of DXL in their own environment, proving they are useful and correct for the intended purpose.

Also, when adding a DXL menu item, doing a one-time check that the right DXL script/behavior is associated with the menu item can increase the detection probability of the “Wrong DXL” error.