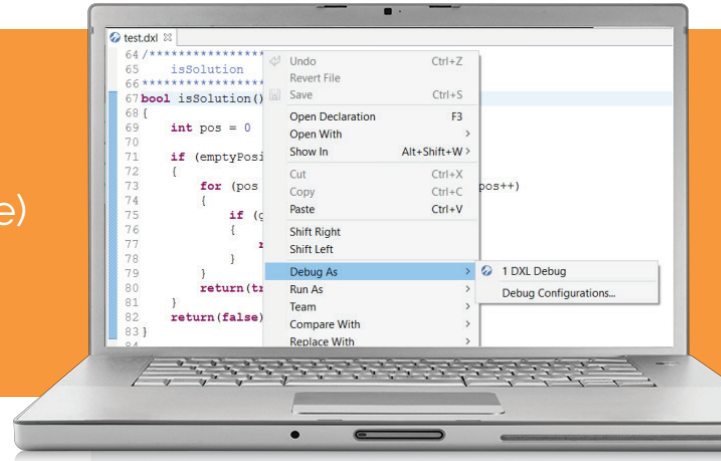


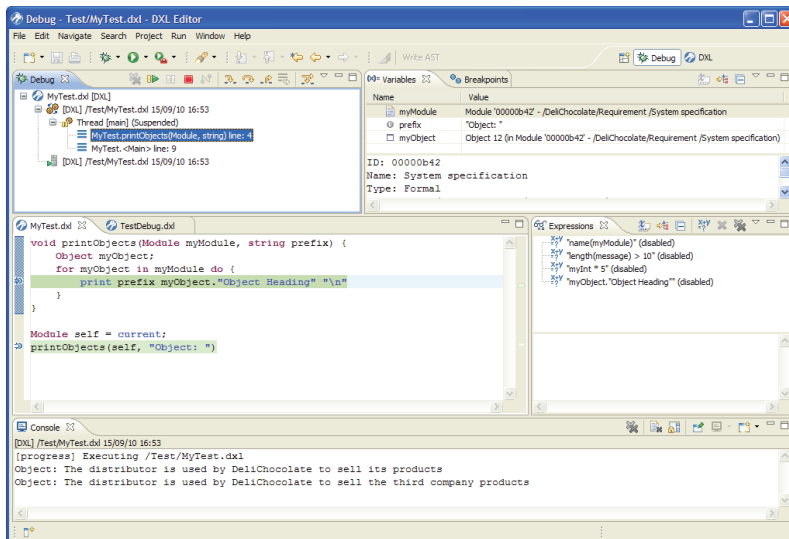
Edit, run, and debug your IBM Rational DOORS DXL (DOORS eXtension Language) scripts from your favorite Eclipse IDE.



IBM Rational DOORS® is a well established platform for engineers, allowing them to manage their project requirements in a flexible environment. DOORS data is manipulated with a dedicated language, DXL (the DOORS eXtension Language).

To overcome the limitations of the DOORS native editor, Sodius has developed the DXL Editor, offering unmatched features to facilitate developers' lives.

Going far beyond syntax highlighting, the DXL Editor is a real development environment for DXL built on the market-leading Eclipse platform, bringing its richness and power to provide a first-class environment for editing, executing and debugging your DXL scripts, on par with other well known languages such as Java and C++.



Improve efficiency  
& reduce development  
time with DXL Editor.

Features include:

### EDITING

- ✓ Syntax highlight
- ✓ Mark occurrences
- ✓ Content assist
- ✓ Text hover

### BROWSING

- ✓ Outline
- ✓ Project explorer

### COMPILATION

- ✓ Compilation
- ✓ Problems view

### EXECUTION

- ✓ Run
- ✓ Console

### DEBUG

- ✓ Breakpoints insertion
- ✓ Step-by-step debug
- ✓ Variable inspection
- ✓ Expression evaluation

### REFERENCE MANUAL

- ✓ Added to Eclipse help view
- ✓ Copy/paste code





## Feature-rich platform for faster, better results

### Compilation Features

#### COMPILATION

Each time a DXL file is modified and saved, the DXL compiler is automatically executed on that file. For each error, an annotation is added in the editor sidebar, with a tooltip showing the error message, and the corresponding line is underlined.

#### PROBLEMS VIEW

Navigation through DXL errors for a particular file, a project or the entire workspace is done using the Problems view.

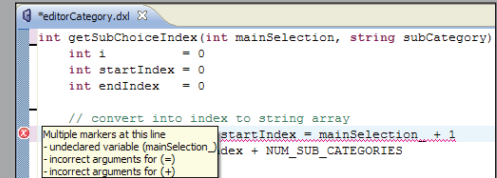
### Execution Features

#### RUN

A DXL file can be executed directly from Eclipse.

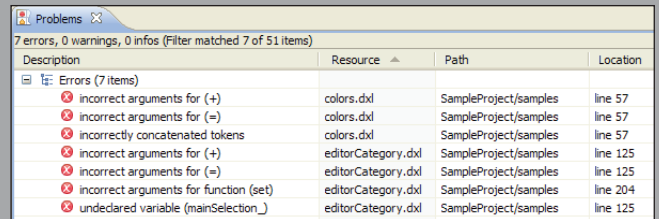
#### CONSOLE

A console is opened when a DXL file is executed. This console logs any DXL text output, and indicates any DXL windows that may be opened by the executed file. Execution errors are displayed in the console and hyperlinks are available on line numbers to jump to the corresponding location in the DXL editor.

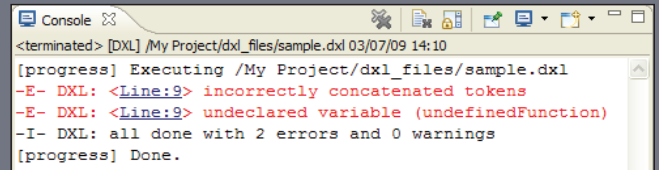


```
int getSubChoiceIndex(int mainSelection, string subCategory)
{
    int i = 0
    int startIndex = 0
    int endIndex = 0

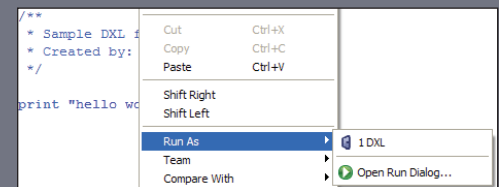
    // convert into index to string array
    startIndex = mainSelection_ + 1
    index + NUM_SUB_CATEGORIES
}
```



Description	Resource	Path	Location
Incorrect arguments for (+)	colors.dxl	SampleProject/samples	line 57
incorrect arguments for (=)	colors.dxl	SampleProject/samples	line 57
incorrectly concatenated tokens	colors.dxl	SampleProject/samples	line 57
incorrect arguments for (+)	editorCategory.dxl	SampleProject/samples	line 125
incorrect arguments for (=)	editorCategory.dxl	SampleProject/samples	line 125
incorrect arguments for function (set)	editorCategory.dxl	SampleProject/samples	line 204
undeclared variable (mainSelection_)	editorCategory.dxl	SampleProject/samples	line 125



```
<terminated> [DXL] /My Project/dxl_files/sample.dxl 03/07/09 14:10
[progress] Executing /My Project/dxl_files/sample.dxl
-E- DXL: <Line:9> incorrectly concatenated tokens
-E- DXL: <Line:9> undeclared variable (undefinedFunction)
-I- DXL: all done with 2 errors and 0 warnings
[progress] Done.
```



## Other DOORS-related products

### MDConnect

Get DOORS database navigation in an Eclipse environment. The RSA Extension interconnects DOORS and Rational Software Architect, facilitating the setup and maintenance of links between requirements and models.

- ✓ Trace Eclipse designs to DOORS requirements
- ✓ Built-in RSA traceability & analysis tool
- ✓ Produce traceability reports from Eclipse

### MDWorkbench

MDWorkbench for DOORS brings advanced functionality that helps to architect DOORS databases, to reverse engineer DOORS schemas, to generate documents and to exchange any DOORS information with other environments.

- ✓ Leverage, convert & transform data across tools
- ✓ Code complex transformation rules in Java
- ✓ Leverage models across tools