

Telelogic 2003 User Group Conference - Americas & Asia/Pacific

Las Vegas - Oct 26-29, 2003

*Work Smarter*

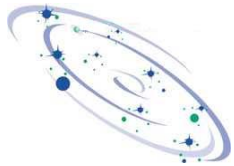
# DOORS Rich Text - Behind the Markup

Michael Sutherland

Galactic Solutions Group LLC

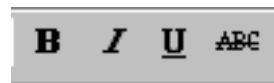
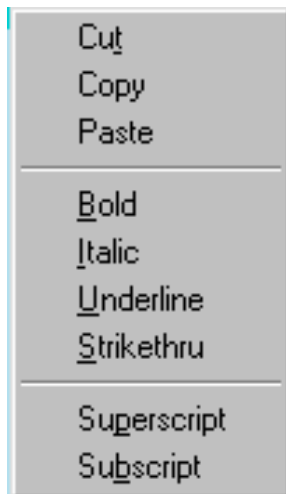
[michael.sutherland@galactic-solutions.com](mailto:michael.sutherland@galactic-solutions.com)





# DOORS 4.0 - Introduction of Rich Text

- With the release of DOORS 4.0, QSS (now Telelogic) introduced Rich Text editing



ID	
1	<b>Bold</b>
2	<i>Italic</i>
3	<u>Underline</u>
4	<del>Strikethru</del>
5	Super <sup>Script</sup>
6	Sub <sub>script</sub>

**Ctrl+Shift+B**

**Makes current selection bold**

**Ctrl+Shift+I**

**Makes current selection italic**

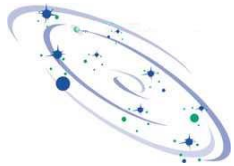
**Ctrl+Shift+U**

**Makes current selection underlined**

**Ctrl+Shift+S**

**Makes current selection strikethru**

Note: No key combinations are available for superscript and subscript.



# Rich Text Markup - Tags

- In DOORS 4, support for simple Rich Text Markup such as font style and font effects (bold, italic, etc.) was added.

	Documented Markup Tag Strings	
Font Effect	On	Off
<b>Bold</b>	"{\\b "	"}"
<i>Italic</i>	"{\\i "	"}"
<u>Underline</u>	"{\\ul "	"}"
<del>Strikethru</del>	"{\\strike "	"}"
Super <sup>script</sup>	"{\\super "	"}"
Sub <sub>script</sub>	"{\\sub "	"}"
	"{\\nosupersub "	"}"



# Rich Text Markup - Syntax

---

The syntax for using these tags within a string is as follows:

`{tag<space>text}` or `{tag{text}}`

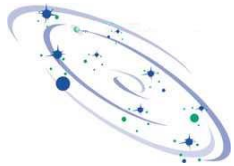
Example: Create the following formatted text:

**Bold** *Italic* ***BoldandItalic***

The following examples shows Rich Text Markup strings that will achieve the desired result:

`"{\b Bold} {i Italic} {\b {i BoldandItalic}}"`

`"{\b Bold} {i Italic} {\b i BoldandItalic}"`



# Setting Rich Text via DOORS GUI v5.2

- DOORS "Object Text" with Rich Text Markup, set by user

ID	
1	<b><i>BoldandItalic</i></b>

- DXL to show Rich Text Markup string for DOORS "Object Text"

The screenshot shows a window titled "DXL Interaction - DOORS". It contains two text areas: "DXL input" and "DXL output".

**DXL input:**

```
string s = ""  
  
Object o = current Object  
s = richText( o."Object Text" )|  
print s
```

**DXL output:**

```
{\b Bold} {\i Italic} {\b {\i BoldandItalic}}
```

At the bottom of the window are buttons for "Save As...", "Browse...", "Close", and "Help". To the right of the window, a "Version:" label is followed by a text box containing "DOORS 5.2".



# Setting Rich Text via DOORS DXL v5.2

- DXL to set DOORS "Object Text" with a Rich Text string

The screenshot shows a window titled "DXL Interaction - DOORS". It contains two main text areas: "DXL input" and "DXL output".

**DXL input:**

```
string s = "{\b Bold} {\i Italic} {\b {\i BoldandItalic}}"
```

```
Object o = current Object
o."Object Text" = richText s
s = richText( o."Object Text" )
print s
```

**DXL output:**

```
{\b Bold} {\i Italic} {\b {\i BoldandItalic}}
```

At the bottom of the window, there is a toolbar with buttons for "Next error", "Print...", "Load...", "Save As...", "Browse...", "Close", and "Help".

- Resulting DOORS "Object Text" with Rich Text Markup

ID	
1	<b><i>BoldandItalic</i></b>

Version: DOORS 5.2



# DOORS 6.0 - Major Rich Text Update

---

- With the release of DOORS 6.0, Telelogic adopted the Microsoft Rich Text Format (RTF) Specification (or some reasonable subset thereof).
- The Microsoft Rich Text Format is a metafile standard developed by Microsoft to encode formatted text and graphics for transfer between applications.

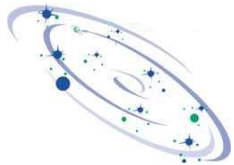
## **Rich Text Format (RTF) Specification, version 1.6**

Microsoft Corporation

May 1999

**Summary:** The Rich Text Format (RTF) Specification provides a format for text and graphics interchange that can be used with different output devices, operating environments, and operating systems. RTF uses the American National Standards Institute (ANSI), PC-8, Macintosh, or IBM PC character set to control the representation and formatting of a document, both on the screen and in print. With the RTF Specification, documents created under different operating systems and with different software applications can be transferred between those operating systems and applications. (248 printed pages)

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnrtfspec/html/rftspec.asp>



# Setting Rich Text via DOORS DXL v6.0

- DXL to set DOORS "Object Text" with a Rich Text Markup string

The screenshot shows a window titled "DXL Interaction - DOORS". It contains two text areas: "DXL input" and "DXL output".

**DXL input:**

```
string s = "{\b Bold} {\i Italic} {\b {\i BoldandItalic}}"  
  
Object o = current Object  
o."Object Text" = richText s  
s = richText( o."Object Text" )  
print s
```

**DXL output:**

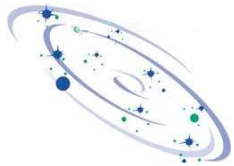
```
{\b Bold} {\i Italic} {\b {\i BoldandItalic}}
```

At the bottom of the window, there are several buttons: "Next error", "Print...", "Load...", "Save As...", "Browse...", "Close", and "Help".

- Resulting DOORS "Object Text" with Rich Text Markup

ID	
1	<b><i>Bold Italic BoldandItalic</i></b>

Version: DOORS 6.0 SR1



# Printing Rich Text Strings DOORS v6.0

Version: DOORS 5.2

DXL output

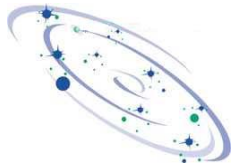
```
{\b Bold} {\i Italic} {\b {\i BoldandItalic}}
```

Version: DOORS 6.0 SR1

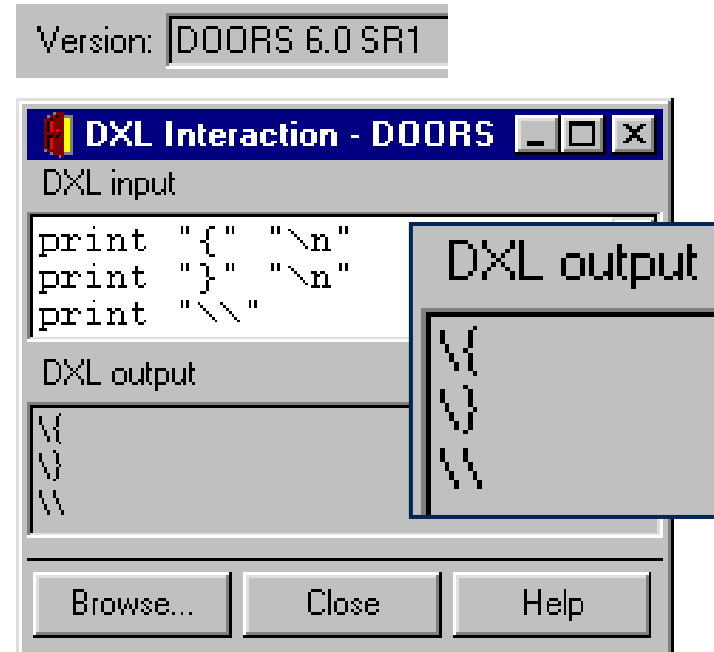
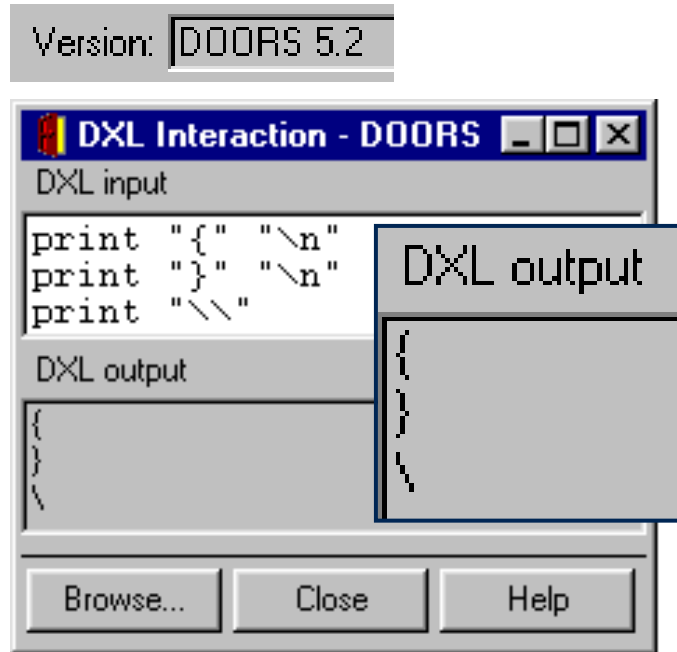
DXL output

```
\{\b Bold\} \{\i Italic\} \{\b \{\i BoldandItalic\}\}
```

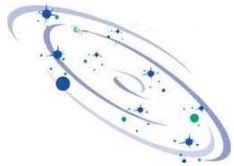
- Note extra "`\`" characters in the version of the Rich Text Markup string printed from DXL.
- In DOORS 6, printing the opening or closing brace characters ( "`{`" or "`}`" ) or a backslash "`\`" results in an extra backslash character being displayed in the printed output.



# Printing Rich Text Strings DOORS v6.0



- Note extra " \" characters in the version of the strings printed from DXL.
- This is a known bug in DOORS 6.x that has been fixed in DOORS 7.



# Setting Rich Text via DOORS GUI v6.0

- DOORS "Object Text" with Rich Text Markup, set by user

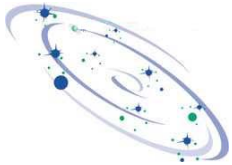
ID	
1	<b><i>Bold Italic BoldandItalic</i></b>

- DXL to show Rich Text Markup string for DOORS "Object Text"

```
DXL Interaction - DOORS
DXL input
string s = ""
Object o = current Object
s = richText( o."Object Text" )
print s
DXL output
\\rtf1\\ansi\\ansicpg1252\\deff0\\deflang1033\\fonttbl{\\f0\\fnil\\prq1\\charset0 Times New Roman;}
\\colorbl ,\\red0\\green0\\blue0;} \\viewkind4\\uc1\\pard\\cf1\\b\\f0\\fs20 Bold\\b0 \\i Italic\\i0 \\b\\i
BoldandItalic\\par \\
```

Version: **DOORS 6.0 SR1**

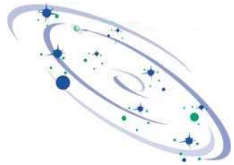
```
DXL output
\\rtf1\\ansi\\ansicpg1252\\deff0\\deflang1033\\fonttbl{\\f0\\fnil\\prq1\\charset0 Times New Roman;}
\\colorbl ,\\red0\\green0\\blue0;} \\viewkind4\\uc1\\pard\\cf1\\b\\f0\\fs20 Bold\\b0 \\i Italic\\i0 \\b\\i
BoldandItalic\\par \\
```



# Default Markup - DOORS 6.0

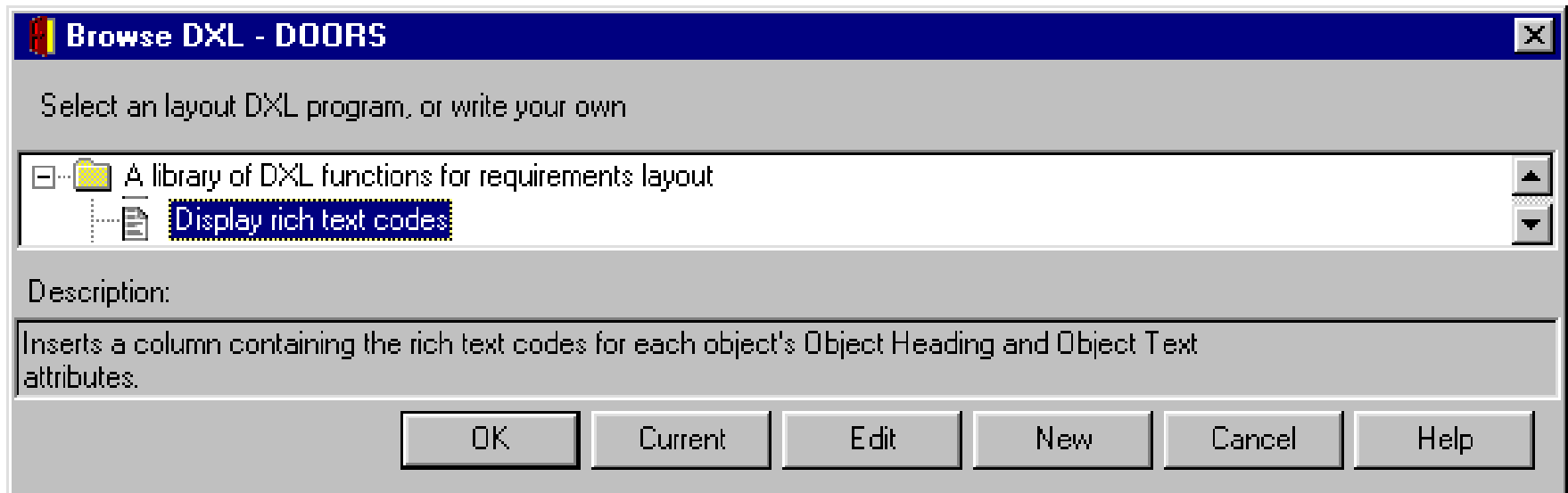
---

- The encoded data is called a RTF *code stream*, which consists of fields called *control words* (formatting and printing instructions), *control symbols* (escape character sequences), and *groups* (enclosed in brace ({} ) characters). Each field begins with a backslash (\) character.
- Object Text - Default Markup (for non-empty Object Text)
- ```
\{\rtf1\ansi\ansicpg1252\deff0\deflang1033
\fonttbl{\f0\fnil\prq1\charset0 Times
New Roman;}{\f1\fnil\prq1\charset0
Century Schoolbook;} \colortbl
;\red0\green0\blue0;}
\viewkind4\uc1\pard\cf1\f0\fs20 TEXT
\f1\par }
```

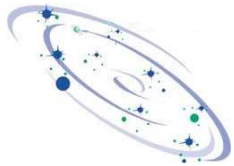


# Viewing Rich Text Markup

- A Layout DXL called “Display rich text codes” (“obj\_richtext.inc”) is available that will display the Rich Text Markup for the DOORS “Object Heading” and “Object Text” attributes.



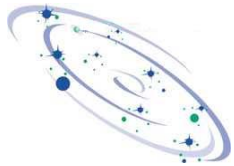
```
$DOORSHOME/lib/dxl/layout/actual/obj_richtext.inc
```



# Font Effects in DOORS 6

## Bold, Italic, Underline

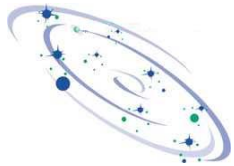
| ID |                  | DOORS 6 Rich Text Format Markup                                                                                                                                                                                                                     |
|----|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | <b>Bold</b>      | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\fprq1\fcharset0 Times New Roman;}{\f1\fnil\fprq1\fcharset0 Century Schoolbook;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\cf1\ub\f0\fs20 Bold\b0\f1\par}</pre>           |
| 2  | <i>Italic</i>    | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\fprq1\fcharset0 Times New Roman;}{\f1\fnil\fprq1\fcharset0 Century Schoolbook;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\cf1\ui\f0\fs20 Italic\i0\f1\par}</pre>         |
| 3  | <u>Underline</u> | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\fprq1\fcharset0 Times New Roman;}{\f1\fnil\fprq1\fcharset0 Century Schoolbook;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\cf1\uul\f0\fs20 Underline\u\none\f1\par}</pre> |



# Font Effects in DOORS 6

## Strikethru, Superscript, Subscript

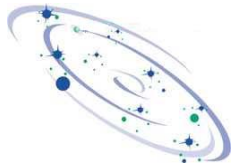
| ID |                         | DOORS 6 Rich Text Format Markup                                                                                                                                                                                                                               |
|----|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4  | Strikethru              | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\fprq1\fcharset0 Times New Roman;}{\f1\fnil\fprq1\fcharset0 Century Schoolbook;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\cf1\strike\f0\fs20 Strikethru\strike0\f1\par}</pre>      |
| 5  | Super <sup>Script</sup> | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\fprq1\fcharset0 Times New Roman;}{\f1\fnil\fcharset0 Century Schoolbook;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\cf1\f0\fs20 Super\super Script\nosupersub \super\f1\par}</pre> |
| 6  | Subscript               | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\fprq1\fcharset0 Times New Roman;}{\f1\fnil\fcharset0 Century Schoolbook;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\cf1\f0\fs20 Sub\sub Script\nosupersub \sub\f1\par}</pre>       |



# New Font Effects in DOORS 6

## Bullet, Indent

| ID |          | DOORS 6 Rich Text Format Markup                                                                                                                                                                                                                                                                                                    |
|----|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7  | • Bullet | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\prq1\fcharset0 Times New Roman;}{\f1\fnil\prq1\fcharset0 Century Schoolbook;}{\f2\fnil \fcharset2 Symbol;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard{\pntext \f2\B7\tab}{\*\pn\pnlvlblt\pnf2\pnindent0 {\pntxtb \B7}}\cf1\f0\fs20 Bullet\f1\par }</pre> |
| 8  | Indent   | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\prq1\fcharset0 Times New Roman;}{\f1\fnil\prq1\fcharset0 Century Schoolbook;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\li360\cf1\f0\fs20 Indent\f1\par }</pre>                                                                                         |
| 9  | Indent   | <pre>{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\fnil\prq1\fcharset0 Times New Roman;}{\f1\fnil\prq1\fcharset0 Century Schoolbook;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\li720\cf1\f0\fs20 Indent\f1\par }</pre>                                                                                         |



# DOORS 6 - Bullets and Indents

- The `\liN` control word specifies a left indent level, in Twips. A Twip is twentieth of a Point (1/1440th of an inch or 1/567th of a centimeter).
- In DOORS 6.0 and above, the base unit of indentation is 360 twips, so values of the left indent level will be multiples of 360.
- In DOORS 7, the `applyTextFormattingToParagraph()` DXL function was introduced to allow the application of bullets and indents to paragraphs through a single DXL command. In DOORS 6, the only way to programmatically add bullets and indents is to comprehend and apply detailed knowledge of Microsoft Rich Text Markup and string manipulation.

Version : DOORS 7.0

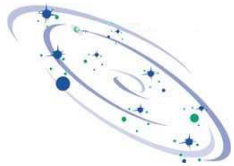
```
applyTextFormattingToParagraph
```

```
string applyTextFormattingToParagraph(string s, bool  
addBullets, int indentLevel, int paraNumber)
```



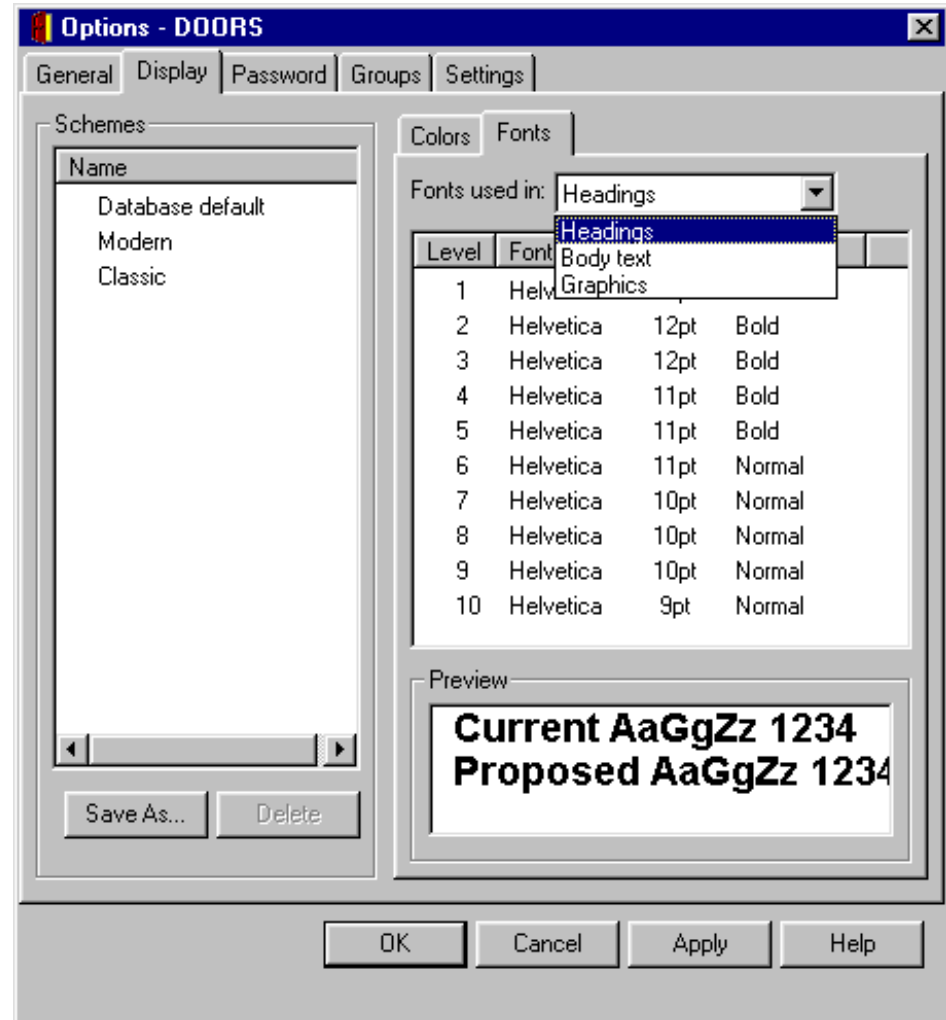
# DOORS 6 - Markup for OLE Objects

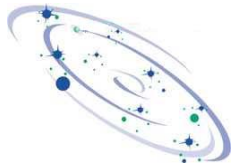
| ID | DOORS 6 Rich Text Format Markup                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10 | <p>?&lt; {\\rtf1\\ansi\\ansicpg1252\\deff0\\deflang1033{\\fonttbl{\\f0\\fnil\\fprq1\\fcharset0 Times New Roman;}<br/>{\\f1\\fnil\\fcharset0 Century Schoolbook;}}<br/>{\\colortbl ;\\red0\\green0\\blue0;}<br/>\\viewkind4\\uc1\\pard\\cf1\\f0\\fs20{\\object\\objemb{\\*\\objclass Paint.Picture}\\objw240<br/>\\objh200{\\*\\objdata<br/>01050000<br/>02000000<br/>07000000<br/>50427275736800<br/>00000000<br/>00000000<br/>c0040000<br/>424dae0400000000000000360400002800000000c0000000a00000001000800000000007800000012<br/>...<br/>01050000<br/>00000000<br/>}}{\\result{\\pict\\wmetafile8\\picw423\\pich353\\piewgoal240\\pichgoal200<br/>010009000003b504000002005e02000000000500000000b02000000000500000000c026101a70105<br/>...<br/>}}\\f1\\par<br/>}</p> |



# DOORS - Font Size

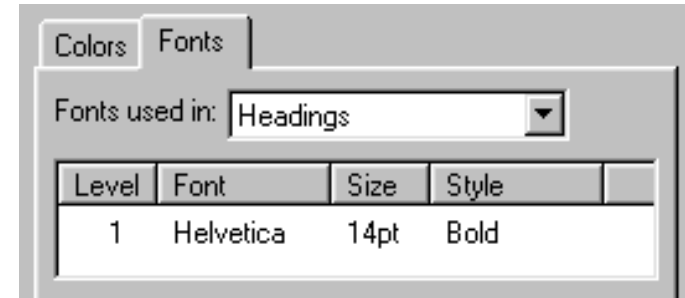
- Any markup relating to Font Size (`\fsN`) is ignored.
- DOORS controls Font Size for Headings and Body text for each individual user through “Display Scheme” settings available from the Tools -> Options menu on the DOORS Database Explorer.



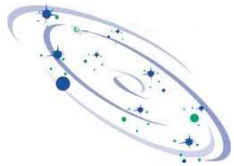


# DOORS - Object Headings

- Text for “Object Heading” is shown as Bold if the Style “Bold” is set for the Display Scheme
- Additional markup may override this setting, making the text appear unbolded



| ID |                               |                                                                                                                                                                                                                         |
|----|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | <b>1 This is Heading Text</b> | Object Heading: {\rtf1\ansi\ansicpg1252\deff0\deflang1033<br>\fonttbl{\f0\fnil\fprq1\fcharset0 Arial;}}<br>\colortbl ; \red0\green0\blue0;}<br>\viewkind4\uc1\pard\cf1\f0\fs28 This is Heading Text\par<br>}            |
| 2  | <b>2 This is Heading Text</b> | Object Heading: {\rtf1\ansi\ansicpg1252\deff0\deflang1033<br>\fonttbl{\f0\fnil\fprq1\fcharset0 Arial;}}<br>\colortbl ; \red0\green0\blue0;}<br>\viewkind4\uc1\pard\cf1\f0\fs28 This is Heading Text <b>\b</b> \par<br>} |



# Application: Layout DXL Markup

- Example: Take the output from the DOORS Analysis->Wizard and add Rich Text bolding to the Attribute names.

```
Edit Layout DXL - DOORS
if (isDeleted othero) continue
otherMod = module othero
doneOne = true
if (depth == 1) {
  s = (identifier othero)
  s = "Object Identifier:" s
  displayRich s
  s = probeRichAttr_(othero,"Object Text", false)
  s = "Object Text:" s
  displayRich s
}
```

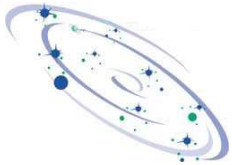
```
Edit Layout DXL - DOORS
if (isDeleted othero) continue
otherMod = module othero
doneOne = true
if (depth == 1) {
  s = (identifier othero)
  s = "{\b " "Object Identifier:" "}" s
  displayRich s
  s = probeRichAttr_(othero,"Object Text", false)
  s = "{\b " "Object Text:" "}" s
  displayRich s
}
```

Out-links

Object Identifier:SR-111  
Object Text:The car shall be powered by a petrol engine if research proves this to be economical.

Out-links

**Object Identifier:** SR-111  
**Object Text:** The car shall be powered by a petrol engine if research proves this to be economical.



# Application: History Markup

- A “histcol.dxl” script is included with DOORS 6.0 that uses Rich Text Markup to show changes to “Object Text” with respect to some original version (from a Baseline, etc.).

**Insert dynamic history column - DOORS**

Instructions:

This tool creates a DXL column or attribute which uses a rich-text markup scheme to show changes to Object Text with respect to some original version.  
The original text is stored in a text attribute called "Old Text", and

Source of original version:

From current

From baseline Baseline: 2.1 (2003)

From attribute Attribute:

Fonts to use for marking text:

Inserted text:  Bold  Italic  Underline  StrikeThrough

Deleted text:  Bold  Italic  Underline  StrikeThrough

Create:  DXL column  DXL attribute

Maximum size of text to compare: 1000

Create Close

`$DOORSHOME/lib/dxl/example/histcol.dxl`

DOORS Rich Text - Behind the Markup v1.1

© 2003 Galactic Solutions Group LLC - Michael Sutherland - michael.sutherland@galactic-solutions.com



# Application: History Markup

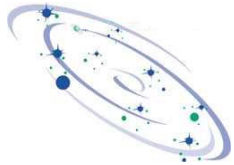
## 'User Requirements' baseline 2.1 (2003)

| Object Iden | User requirements for passenger car                                                                          |
|-------------|--------------------------------------------------------------------------------------------------------------|
| UR359       | This module contains the user requirements for a new car to be commercially available by 1st September 2002. |

## User Requirements' current 2.1 (2003)

| Object Iden | User requirements for passenger car                                                                         | Dynamic history 2.1 (2003)                                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| UR359       | This module contains the user requirements for a new car to be commercially available by 27th October 2003. | This module contains the user requirements for a new car to be commercially available by <del>1st 27th September 2002, October 2003.</del> |

| Dynamic history 2.1 (2003)                                                                                                                                                                                                |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| This module contains the user requirements for a new car to be commercially available by <del>1</del> <del>st</del> <del>27</del> <del>th</del> <del>September</del> <del>2002.</del> <del>October</del> <del>2003.</del> |



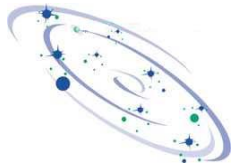
# Application: Rich Text “Diffing”

- The “`histcol.dxl`” script included with DOORS 6.0 uses the following DXL library to generate a Rich Text string describing the differences between two strings. New words are underlined, and deleted words are scored through.
- `#include <example/include/delta.inc>`
- In DOORS 7.0, a built in DXL function “`diff()`” was introduced, which computes the annotated difference, or “redlined difference” between source and target, using markup and color.

```
diff(buffer)
```

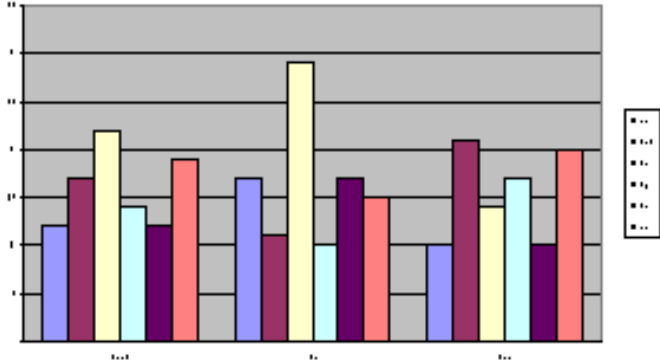
```
string diff(Buffer result, source, target, string  
removeMarkup, insertMarkup)
```

```
string diff(Buffer result, source, target)
```



# Application: Rich Text Traceability

- Example: When using the Analysis -> Wizard to create Layout DXL Column, make sure "Include OLE objects in text" is selected in Step 5.

| User requirements for passenger car                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Out-links                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>While walking through the use of the wizard you will need to refer to this paragraph, so be sure you can switch back and forth between the wizard and this window. Start the Analysis Wizard from the <i>Analysis</i> menu and press next to get to the first functional window.</p> <p>On this window of the wizard you can select which other documents you want to include in the analysis. Click on "All modules" option for both the modules and link modules options (not "All open modules" ) On the second window select "Out-links" as the links we are looking for go from this document out to the System Requirements Document. On the third window remove the "Object Heading" from the list of attributes to display and select "Object Text" instead (NOT "Object Short Text" - all the links you want to see are to text objects, not headings.) On the fourth window turn off the recursive analysis option (you only want to trace down one level) and turn on the option to show attribute names. Leave the option for one attribute per line turned on. Press "Next" one more time and you are finished; a column will be generated showing linked data.</p> <p>Make this text column smaller and the new column wider by dragging the borders as you did before. <b>Scroll down until you find the next linked object (UR 17) and read that text.</b></p> | <p>System Requirements<br/>SR-111</p> <p>The car shall be <b>powered</b> by a petrol engine if research proves this to be economical.</p>  |



# DOORS Module Font Table

- When information from Microsoft Word or RTF documents is imported into a DOORS Module, a Module level attribute called "Font Table" may be created and populated with non-default font table entries. The DOORS DXL manual states that these entries are “used for mapping richtext font markup to character set information to override embedded font markup.”

## Example

```
print fontTable current Module
```

In a newly created module, the DOORS default font table is:

```
{\f1012\fwiss\fcharset177 Arial;}{\f1011\fwiss\fcharset162  
Arial;}{\f1010\fwiss\fcharset238 Arial;}{\f1009\fwiss\fcharset204  
Arial;}{\f1008\fwiss\fcharset161 Arial;}{\f1007\fwiss\fcharset0  
Arial;}{\f1006\froman\fcharset177 Times New  
Roman;}{\f1005\froman\fcharset162 Times New  
Roman;}{\f1004\froman\fcharset238 Times New  
Roman;}{\f1003\froman\fcharset204 Times New  
Roman;}{\f1002\froman\fcharset161 Times New Roman;}{\f1001\ftch\fcharset2  
Symbol;}{\f1000\froman\fcharset0 Times New Roman;}
```



# Rich Text Normalization in DOORS 6

---

- The DOORS 5 Manual states: "In versions of DOORS prior to 4.1, the DOORS import tools imported rich text markup that wasn't supported by DOORS. DOORS now only imports supported rich-text markup, and discards markup that it doesn't support."
- Now that DOORS 6 and above has implemented the Microsoft Rich Text Format (RTF) standard more fully, some of this markup that was previously suppressed may reemerge when DOORS 5 data is converted to DOORS 6. In particular, font markup may appear to be "non-standard" with regards to Fonts.



# Rich Text Normalization - Solutions

---

- The solutions recommended by Telelogic are:
  - (1) "Remove Unsupported Markup" (attempt to remove only unsupported markup)
  - (2) "Remove All Markup" (strip out all Rich Text Formatting, leaving raw text with no formatting)
- These functions first provided in DOORS 4.1 and available in DOORS 5 were not included with DOORS 6. Attempting to use them results in less than desirable results. Solutions (1) and (2) do not preserve Symbol Font characters, and leave unwanted Font Style information. Solution (2) is not acceptable because valid markup must be preserved.
- In DOORS 5, deleting the contents of the Module level "Font Table" attribute eliminated some non-standard fonts from appearing. In DOORS 6, this does not seem to have the same effect.



# Rich Text Normalization - Requirements

---

The requirements for the desired normalization are:

- All Font Markup shall be eliminated for Object Heading and Object Text Attributes, except Symbol Font characters, which shall be preserved.
- Valid Rich Text Markup (bold, italic, strikethru, underline, superscript, subscript) shall be preserved.
- Object Text with embedded OLE, bullet, or indent markup (new features in DOORS 6) shall not be modified.



# Rich Text Normalization - Algorithm

---

- Use a "for rt in s" loop to scan the existing Rich Text string "chunk" by "chunk" .
- Use "rt.bold", "rt.italic", etc. to detect valid markup for each "chunk". This method has the benefit of ignoring any invalid markup.
- A new version of the Rich Text string is built by taking each successive "chunk" of formatting and reapplying the valid markup to the plain text version of the string.



# Rich Text Normalization - Algorithm

```
// s is Rich Text string
string s = richText( o."Object Text" )

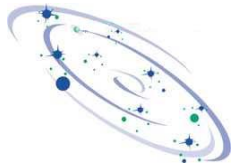
Buffer richBufferNew = create

RichText rt
for rt in s do {
    string text = rt.text
    string markupPrefix = ""
    string markupSuffix = ""

    if ( rt.bold ) {
        markupPrefix = "{\\b " markupPrefix
        markupSuffix = markupSuffix "}"
    }
    if ( rt.italic ) {
        markupPrefix = "{\\i " markupPrefix
        markupSuffix = markupSuffix "}"
    }
    ...
    if ( rt.charset == charsetSymbol ) {
        markupPrefix = markupPrefix "{\\f1001 "
        markupSuffix = "}" markupSuffix
    }

    // Apply markup prefix and suffix to plain text
    richBufferNew += markupPrefix text markupSuffix
}
```

The code is from a free DXL utility called "[Normalize Rich Text Markup](http://galactic-solutions.com)", available from <http://galactic-solutions.com>.



# Rich Text Normalization - Implication

DOORS 5 - Example of "for rt in s" loop

Version: DOORS 5.2

The screenshot shows a window titled "DXL Interaction - DOORS" with a standard Windows-style title bar. The window is divided into two main sections: "DXL input" and "DXL output".

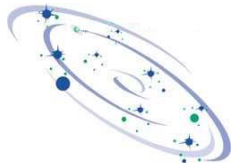
**DXL input:**

```
string s = "{\b Bold} {\i Italic} {\b {\i BoldandItalic}}"  
  
RichText rt  
for rt in s do {  
    if (rt.bold) print "(bold)"  
    if (rt.italic) print "(italic)"  
    print "[" rt.text "]"  
}
```

**DXL output:**

```
(bold)[Bold]  
[ ]  
(italic)[Italic]  
[ ]  
(bold)(italic)[BoldandItalic]
```

Below the output sections, there is a row of buttons: "Next error", "Print...", "Load...", "Save As...", "Browse...", "Close", and "Help".



# Rich Text Normalization - Implication

DOORS 6 - Example of "for rt in s" loop

Version:

**DXL Interaction - DOORS**

DXL input

```
string s = "{\b Bold} {\i Italic} {\b {\i BoldandItalic}}"
```

RichText rt

```
for rt in s do {  
  if (rt.bold) print "(bold)"  
  if (rt.italic) print "(italic)"  
  print "[" rt.text "]" \n  
}
```

DXL output

```
(bold)[Bol  
(bold)[d  
[ ]  
(italic)[l  
(italic)[tal  
(italic)[i  
(italic)[c
```

Next error Print... Load... Save As... Browse... Close Help



# Rich Text Normalization - Implication

- Note that DOORS 6.0 detected more chunks of Rich Text Markup than were actually applied.

Version: `DOORS 5.2`

DXL output

```
(bold)[Bold]
[ ]
(italic)[Italic]
[ ]
(bold)(italic)[BoldandItalic]
```

Version: `DOORS 6.0 SR1`

DXL output

```
(bold)[Bol]
(bold)[d]
[ ]
(italic)[l]
(italic)[tal]
(italic)[i]
(italic)[c]
```

- This is a known bug in DOORS 6.x that has been fixed in DOORS 7.

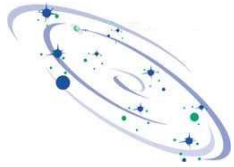
- The "Normalize Rich Text Markup" algorithm will still work with this bug, but it will turn this artificial segmentation into actual segmentation, creating more Rich Text "chunks" than actually necessary.



# Other Topics

---

- The white paper version of this presentation, available from <http://galactic-solutions.com>, covers many of these topics in greater detail, and also covers the following additional topics:
  - *Exporting Rich Text to Microsoft Office Applications*
  - *Microsoft Word - Paragraph vs. Character Style*
  - *Microsoft Word - Updated DOORS 6 Export to Word*
  - *Microsoft Excel - Exporting Rich Text Markup*



# Obtaining Software

---

- DOORS Users are encouraged to obtain, use, share, and improve upon the software mentioned in this presentation.
- For a free copy:

Contact: [michael.sutherland@galactic-solutions.com](mailto:michael.sutherland@galactic-solutions.com)

or download from

<http://galactic-solutions.com>