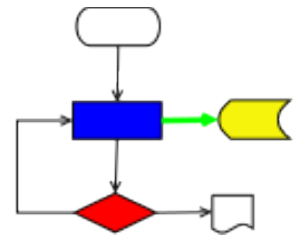


VICI



VISUAL CHART INTERPRETER

VICI

Publication History

Date	Who	What Changes
15 January 2013	Brenton Ross	Initial version.
25 May 2014	Brenton Ross	Updated for VICI



Table of Contents

1 Introduction.....	4
1.1 Scope.....	4
1.2 Overview.....	4
1.3 Audience.....	4
2 XML Catalogue.....	5
3 Vici Command Database.....	6
3.1 Definition.....	6
3.2 Example.....	7
4 Tag Database.....	8
4.1 Definition.....	8
4.2 Example.....	8
5 Script Database.....	9
5.1 Definition.....	9
5.2 Example Script.....	11
Appendix A.....	12

1 Introduction

This is part of the system design for the VICI project.

1.1 Scope

This document covers the design of the persistent storage for the project. This is provided a separate document since it is part of the interface specification between modules.

1.2 Overview

The detailed design includes:

- **Interface Stubs:** A framework of facade classes for the modules.
- **Use Case Descriptions:** A description of how a user is expected to interact with the application.
- **Application Design:** The classes and their relationships.
- **User Interface Design:** The design and layout of the graphical components of the system.
- **Persistent Storage Design:** The specifications for the XML files used to store configuration and scripts. (This document.)

1.3 Audience

This document is intended to be used by the designers and developers, and later the maintainers, of the VICI project.

2 XML Catalogue

The usual method for having an XML file find its corresponding DTD is to include a SYSTEM reference URL in the DOCTYPE element. This works fine if the DTD can be placed in a single well known location, which normally implies somewhere on the web. Of course this doesn't work if the network connection is unavailable, and it makes the development process a bit more complex.

The alternative is to use the PUBLIC identifier in the DOCTYPE element. This identifier is used by the XML library to find the corresponding DTD in a catalogue file that is in turn found via an environment variable: XML_CATALOG_FILES.

This alternative mechanism will be used by VICI.

A typical catalogue file:

```
<?xml version="1.0"?>
<!DOCTYPE catalog
PUBLIC "-//OASIS//DTD Entity Resolution XML Catalog V1.0//EN"
"http://www.oasis-
open.org/committees/entity/release/1.0/catalog.dtd">
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
prefer="public">
  <group xml:base="/home/brenton/dev/gsh/ws/persist/">
    <public publicId="-//VICI/V0.1/command.dtd" uri="command.dtd"/>
  </group>
</catalog>
```

The installation script will need to generate a suitable catalogue file, and set the path to it in the user's environment.

3 Vici Command Database

This is the database of prepared commands. The command database will be in the installation directory so that all users can access it:

/usr/share/vici/command.xml.

3.1 Definition

The following DTD defines the format of the database file which will contain all the command specifications. (The alternative of one file per command would slow down the start up process, and there will only be a few hundred prepared commands.)

```
<!-- DTD for VICI Command Database -->
<!ELEMENT CommandList (Command*) >
<!ELEMENT Command (Name, Description, HelpCmnd*, Options, Alias*) >
<!ELEMENT Name (#PCDATA) >
<!ELEMENT Description (#PCDATA) >
<!ELEMENT HelpCmnd (#PCDATA) >
<!-- A CDATA section will be used to contain the EBNF -->
<!ELEMENT Options (#PCDATA) >
<!ELEMENT Alias (Name, Description, Options) >
<!-- end of DTD for VICI Command Database -->
```

3.2 Example

The following is an example of the Command Database:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CommandList PUBLIC "-//VICI/V0.1/command.dtd" "No System
URI" >
<CommandList>
  <Command>
    <Name>cal</Name>
    <Description>Display a calendar</Description>
    <HelpCmd>man cal</HelpCmd>
    <Options><![CDATA[Options ::= [ Period ] [ DayOption ]
[ Julian ] [ Date ] ;
Date ::= [ [ [ Day ] Month ] Year ] ;
Period ::= Single | Triple | Annual ;
Single ::= "-1" | "--one" ;
Triple ::= "-3" | "--three" ;
Annual ::= "-y" | "--year" ;
DayOption ::= Sunday | Monday ;
Sunday ::= "-s" | "--sunday" ;
Monday ::= "-m" | "--monday" ;
Julian ::= "-j" | "--julian" ;
Day ::= "1" ... "31" ;
Month ::= "1" ... "12" ;
Year ::= "1" ... "9999" ;]]></Options>
  </Command>
</CommandList>
```

4 Tag Database

This database is used to search for commands using key words. The general concept for this is based on the idea of an ontology. The tag database will be in the user's directory: `$HOME/.vici/tags.xml`. It will be initially populated from a copy in `/usr/share/vici`.

4.1 Definition

The DTD for the tag database simply allows commands or tags to be associated with other tags:

```
<!-- DTD for the VICI tags database -->
<!ELEMENT TAGS (ASSOC*) >
<!ELEMENT ASSOC ((Cmd | Tag), (Tag)) >
<!ELEMENT Cmd (#PCDATA) >
<!ELEMENT Tag (#PCDATA) >
<!-- end of DTD -->
```

4.2 Example

The following is an example XML file for a tag database:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TAGS PUBLIC "-//VICI/V0.1/tags.dtd" "No System URI" >
<TAGS>
  <ASSOC><Cmd>cat</Cmd><Tag>text</Tag></ASSOC>
</TAGS>
```


5 Script Database

The script database will be a directory containing multiple XML files, one per script. The scripts will be in a user's directory: \$HOME/.vici/scripts.

5.1 Definition

The DTD for the script:

```
<!-- DTD for VICI Scripts -->

<!-- A script can have multiple functions -->
<!ELEMENT VICI (Signature, User*, Menu, Func*) >
<!ATTLIST VICI
    Version CDATA #REQUIRED
    >

<!ELEMENT Signature (#PCDATA) >

<!ELEMENT User (#PCDATA) >

<!ELEMENT Menu (MenuTitle*) >
<!ELEMENT MenuTitle (MenuItem|MenuTitle)* >
<!ATTLIST MenuTitle Name CDATA #REQUIRED >
<!ELEMENT MenuItem (#PCDATA) >
<!ATTLIST MenuItem Name CDATA #REQUIRED >

<!ELEMENT Func (Shape | Line | Text)* >

<!ELEMENT Shape (Action | Function | Variable | File | IPC) >
<!ATTLIST Shape Ref ID #REQUIRED
    XPos CDATA #REQUIRED
    YPos CDATA #REQUIRED
    Label CDATA #IMPLIED
    >

<!ELEMENT Action (Command) >
<!ATTLIST Action
    Style (Cmnd | Choice | FnRef) #REQUIRED
    BgTask (true | false) #IMPLIED
    >

<!-- empty -->
<!ELEMENT Function (#PCDATA) >
<!ATTLIST Function
    Name CDATA #REQUIRED
    Menu CDATA #IMPLIED
    >

<!-- empty for variables, the value for constants -->
<!ELEMENT Variable (#PCDATA) >
<!ATTLIST Variable
    Name CDATA #REQUIRED
    Style (Var | Const) #REQUIRED
    >

<!-- empty except for inline files -->
<!ELEMENT File (#PCDATA|Label)* >
<!ATTLIST File
```

```

        Path CDATA #IMPLIED
        Style (File | InLine | Temp | Pipe) #REQUIRED
        Append (true | false) #IMPLIED
    >

<!-- empty -->
<!ELEMENT IPC (#PCDATA) >
<!ATTLIST IPC
    Name CDATA #REQUIRED
    Style (Lock | Unlock | Semaphore) #REQUIRED
>

<!ELEMENT Label (#PCDATA) >
<!ATTLIST Label VarRef IDREF #REQUIRED
>

<!-- content is the command options and parameters -->
<!ELEMENT Command (#PCDATA|Label)* >
<!ATTLIST Command
    Name CDATA #REQUIRED
>

<!ELEMENT Line (Point*) >
<!ATTLIST Line From IDREF #REQUIRED
    To IDREF #REQUIRED
    Style (Flow | Success | Fail |
        StdOut | StdErr | StdOutErr |
        Signal ) #REQUIRED
    ExitCode CDATA #IMPLIED
>

<!-- empty -->
<!ELEMENT Point (#PCDATA) >
<!ATTLIST Point
    x CDATA #REQUIRED
    y CDATA #REQUIRED
>

<!ELEMENT Text (#PCDATA) >
<!ATTLIST Text
    Face CDATA #REQUIRED
    Size CDATA #REQUIRED
    Style CDATA #REQUIRED
>

```

5.2 Example Script

The following is the test example, used to verify the DTD:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE VICI PUBLIC "-//VICI/V0.1/script.dtd" "No System URI" >
<VICI Version="0.1">
  <Signature><![CDATA[12345ab56def]]></Signature>
  <User>brenton</User>
  <Menu>
    <MenuTitle Name="Control">
      <MenuItem Name="Run" />
      <MenuItem Name="Pause" />
      <MenuItem Name="Terminate" />
    </MenuTitle>
    <MenuTitle Name="Script">
      <MenuItem Name="Main" />
    </MenuTitle>
  </Menu>
  <Func>
    <Shape Ref="C1" XPos="50" YPos="50" Label="cal">
      <Action Style="Cmnd">
        <Command Name="cal">2012</Command>
      </Action>
    </Shape>
    <Shape Ref="F1" XPos="100" YPos="50" Label="func">
      <Action Style="FnRef" BgTask="true" >
        <Command Name="Func_1"/>
      </Action>
    </Shape>
    <Shape Ref="F2" XPos="150" YPos="50" Label="func2">
      <Action Style="FnRef" BgTask="false" >
        <Command Name="Func_1">test <Label
VarRef="V2"/></Command>
      </Action>
    </Shape>
    <Shape Ref="F3" XPos="200" YPos="50" Label="func">
      <Function Name="func" Menu="Start"/>
    </Shape>
    <Shape Ref="V1" XPos="250" YPos="50" Label="var_1">
      <Variable Name="v1" Style="Var"/>
    </Shape>
    <Shape Ref="V2" XPos="50" YPos="100" Label="const 1" >
      <Variable Name="c1" Style="Const">
        <![CDATA[/home/brenton/dev/gsh]]>
      </Variable>
    </Shape>
    <Shape Ref="D1" XPos="100" YPos="100" Label="file 1" >
      <File Style="File" Append="true"/>
    </Shape>
    <Shape Ref="D2" XPos="100" YPos="100" Label="file 1" >
      <File Style="InLine"><![CDATA[
This is the content of an
inline file with embedded
variables]]><Label VarRef="V2"/></File>
    </Shape>
    <Shape Ref="I1" XPos="100" YPos="150" Label="mutex">
      <IPC Name="mutex_1" Style="Lock"/>
    </Shape>
    <Line From="F2" To="V2" Style="Flow"/>
  </Func>
</VICI>

```

Appendix A