DASSAULT SYSTEMES

ENOVIA Training
Foils

LCA Administration Advanced (3)

Development Environment

Version 5 Release 11
May 2003

EDU-ENOV-E-LAE-AF-V5R11

Course Presentation

Objectives of the Course

In this course, we will learn about the directory tree structure, the specific tools developed, and how to find information in the CAA V5 Encyclopedia.

Targeted audience

ENOVIA V5 Programmers

2 hours

Prerequisites:

CAA V5 Programming

Table of Contents (1/2)

Table of Contents (2/2)

Planning

In this course, you will see the CAA V5 Development Environment

- **A Component Architecture**
- **Compilation Tools**
- **MSDev integration**
- **Customization of ENOVIA LCA**
- **CAA V5 Encyclopedia and Programming Rules**
- **RADE installation and licensing**

A Component Architecture

You will become familiar with CAA V5 Development Environment

- **CAA V5 Development Environment Objectives**
- **CAA V5 Characteristics**
- **Component Application Architecture**
- **CAA V5 Framework**
- **CAA V5 Prerequisite Workspaces**
- **CAA V5 File Tree**
- **Framework Identity Card**

CAA V5 Development Environment Objectives

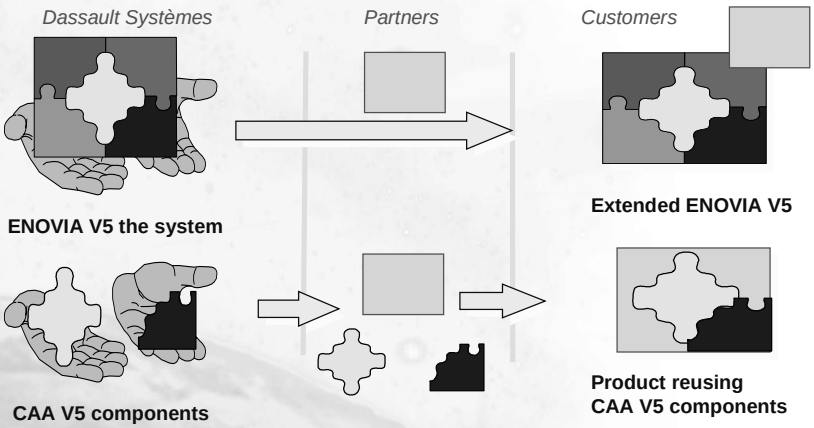
- *Tools and Methods for an OO programming environment*
- *Support the V5 Architecture*
- *Support large teams of developers working concurrently in different sites*
- *Help making better quality software in a faster way*
- *Capture and enforce company processes*

CAA V5 Characteristics

- *Common development platform for all the Dassault Systèmes product lines*
  - *CATIA / ENOVIA / DELMIA*
- *Code written on top of CAA V5 is the same on NT and UNIX*

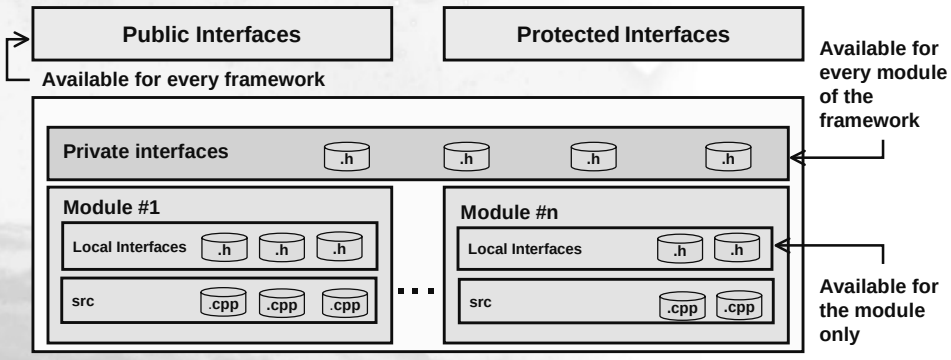
Component Application Architecture

CAA V5 can be used to implement new products

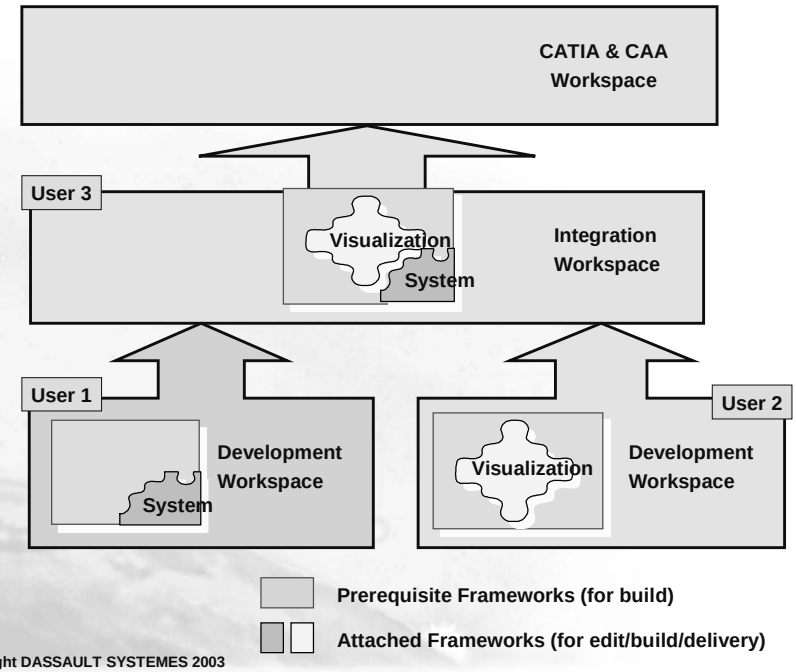


CAA V5 Framework

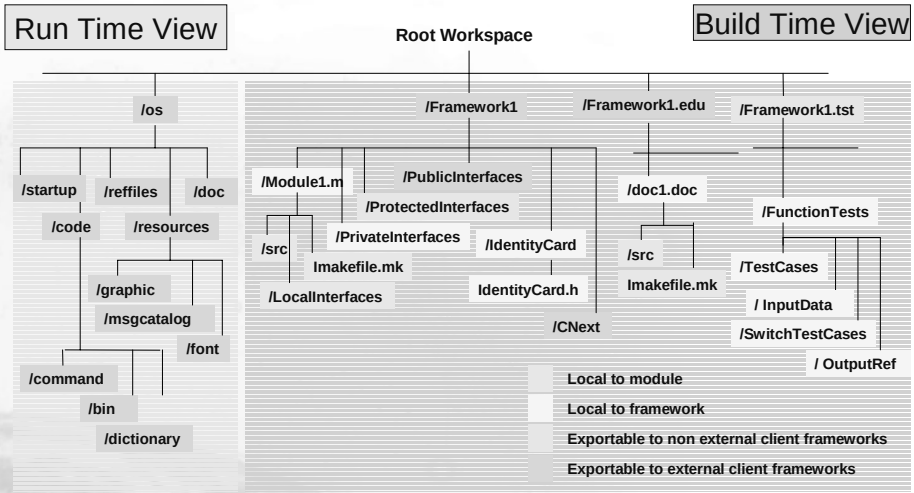
Group of interoperating objects with built-in capabilities delivered as a complete resource to customer applications



CAA V5 Prerequisite Workspaces



CAA V5 File Tree



Framework Identity Card

The IdentityCard defines the prerequisite frameworks to build and use a framework.

- ***One identity card per framework.***
- ***If no prerequisite framework, define an empty IdentityCard.***
- ***This file is used by our building tool to limit the header file search to the corresponding Interface directories of the prerequisite frameworks.***

IdentityCard.h

```
AddPrereqComponent("System",Protected);
AddPrereqComponent("ObjectModelerBase", Protected);
```

This framework uses only headers defined in the PublicInterfaces or ProtectedInterfaces directory of the System and ObjectModelerBase frameworks

Compilation Tools

You will become familiar with CAA V5 Tools

- **Manage the CAA V5 Tool Level : TCK**
- **Define your Prerequisite Workspaces : mkGetPreq**
- **Build your executables : mkmk**
- **mkmk : The Imakefile.mk**
- **Build with external libraries**
- **About mkmk**
- **mkmk Tips**
- **ExportedByModuleName Processor Variables**
- **Runtime Tools**
- **Test Tools : mkodt**

Manage the CAA V5 Tool Level : TCK

- *The Tool Configuration Key manages several levels of the CAA V5 RADE tools.*
- *To set up the tck environment :*
  tck_init
- *To list the different levels available*
  tck_list
- *To set up a specific tool level*
  tck_profile LevelNameYouWantToUse

Define your prerequisite workspaces : mkGetPreq

- *mkGetPreq -p PrerequisiteWorkspace1*
  This enables you to define where the prerequisite resources are located
  Build time: header files
  Run time: shared libraries, resource files …

  This command must be launched in a window where the CAA V5 environment has been set and the current directory is your workspace

Build your executables : mkmk

A unique Dassault Systèmes tool built on top of the standard compilers working in the same way on UNIX and Windows NT:

Compile Fortran, C, C++, IDL, Express, CIRCE, ...

Link-edit

It uses the Imakefile.mk file that must be defined for every module.

mkmk : The Imakefile.mk

Imakefile.mk

```
BUILT_OBJECT_TYPE=SHARED LIBRARY        <-- Define the module type

Define the build options common to all the OS
OS = COMMON
WIZARD_LINK_MODULES =  \                Specific keyword used by the wizards
JS0GROUP JS0FM  CATApplicationFrame     The continuation character is "\"

LINK_WITH = $(WIZARD_LINK_MODULES) \    Defines the shared libraries that
            CATDialogEngine             resolve the symbols you use
OS = AIX
SYS_INCPATH =                           Define the build options specific to a given OS if necessary
SYS_LIBS = -lXm -lXt -lXmu -lX11 -lm
SYS_LIBPATH = -L/usr/lpp/X11/lib/R5/Motif1.2 -L/usr/lpp/X11/Motif1.2/lib
...
```

Build with external libraries

Imakefile.mk

On NT

```
# Link with external libraries
LOCAL_LDFLAGS = /LIBPATH:"E:\DirectoryWhereTheLibrariesAreStored"
# Name of the libraries
SYS_LIBS = LibraryName.lib
# Link with include files
LOCAL_CCFLAGS = /I"E:\DirectoryWhereTheIncludeFilesAreStored"
```

On UNIX

```
# Link with external libraries
LOCAL_LDFLAGS = -L/MachineName/DirectoryWhereTheLibrariesAreStored
# Name of the libraries
SYS_LIBS = LibraryName
# Link with include files
LOCAL_CCFLAGS = -I/MachineName/DirectoryWhereTheIncludeFilesAreStored
```


About mkmk

Its behavior depends on the current directory:

your workspace directory is the current directory

mkmk –aug → to force all the modules to be rebuilt with the debug option.

mkmk –a → to rebuild only what needs to be rebuilt

A module directory is the current directory:

mkmk –ug → to force the corresponding module to be rebuilt with the debug option.

mkmk → to rebuild only if necessary

To access the mkmk Help On Line, use mkmk -h.

mkmk Tips

Use the update (-u) option when:

modifying the dependencies (an include file added or suppressed)

adding or removing a file (.h and .cpp).

modifying the IdentityCard.h and/or the Imakefile.mk

In other cases, do not use the update option. mkmk will reuse some intermediate files generated before.

Objects
ImportedInterfaces
various

ExportedByModuleName Preprocessor Variables

A Windows NT mechanism imposes that shared libraries declare explicitly what they import and export.

To manage this, we define some pre-processor variables.

MyClass.h

```
#include "MyModule.h"
Class ExportedByMyModule MyClass
{
...
```

MyModule.h

```
#ifdef  _WINDOWS_SOURCE
#ifdef  __MyModule
#define ExportedByMyModule __declspec(dllexport)
#else
#define ExportedByMyModule __declspec(dllimport)
#endif
#else
#define ExportedByMyModule
#endif
```

Variable defined by mkmk on Windows NT

Variable defined by mkmk when building MyModule

Runtime tools

mkrtv

copy the application resources (icons, message files, dictionaries, …) from the Build time directories into the Run time directories.

mkrun

run CATIA V5 or any main executable developed on top of CAA V5

mkrun -c MyProgram

Test Tool : mkodt

mkodt

Every framework FW should provide its test framework: FW.tst

Uses some predefined environment variables

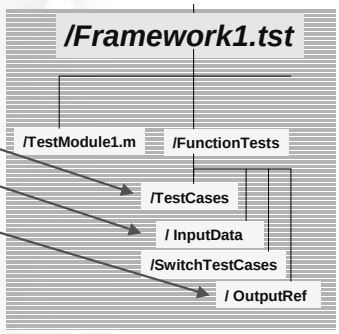
ADL_ODT_OUT,

ADL_ODT_TMP

Contains shells that launch the test programs

Contains any data required by the shells: models, ...

Contains any reference data that can be used by the shells to check what is produced by a test program



MSDev Integration

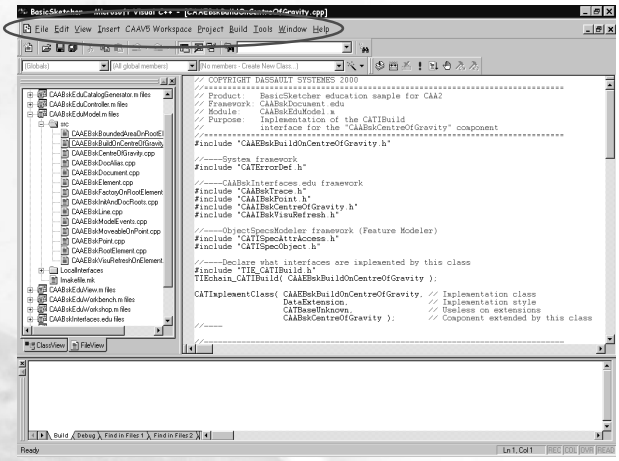
You will learn to use the C++ Developer Studio

- Microsoft Developer Studio CAA V5 Add-Ins
- CAA V5 wizards in Microsoft Developer Studio
- CAA V5 Object Browser
- Mapping between commands an MSDev Add-Ins
- MSDev Add-Ins Hints and Tips
- Enable porting on UNIX Visual C++
- Activate the Porting on UNIX
- Porting on UNIX
- Other Tools used in the CAA V5 context

Microsoft Developer Studio CAA V5 Add-Ins

All our specific tools have been integrated in Microsoft Developer Studio V6

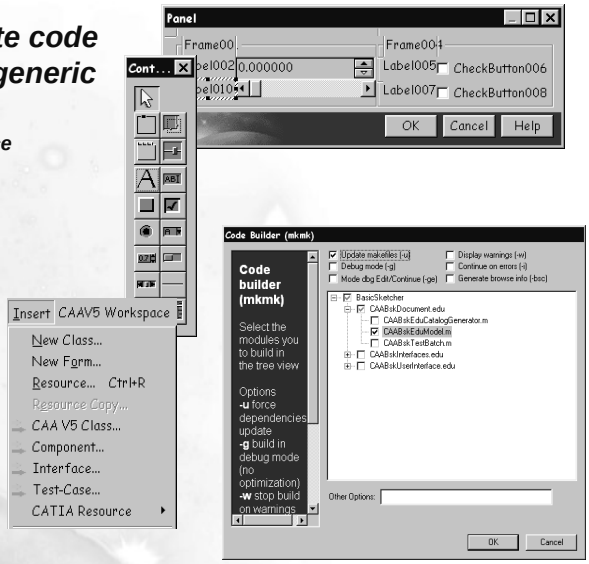
Must be installed using the Unicode String option.



CAA V5 wizards in Microsoft Developer Studio

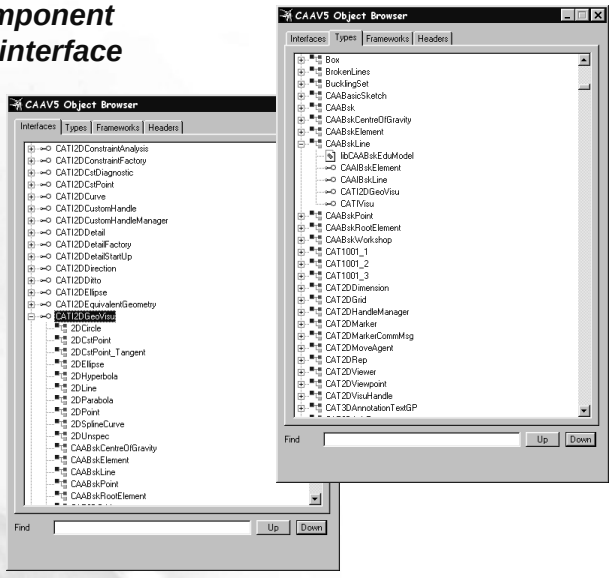
Wizards to generate code corresponding to generic tasks:

- ***New CAA V5 Workspace***
- ***New Framework***
- ***New Module***
- ***New Command***
- ***New Panel***
- ***New Interface***
- ***New implementation***
- ***...***

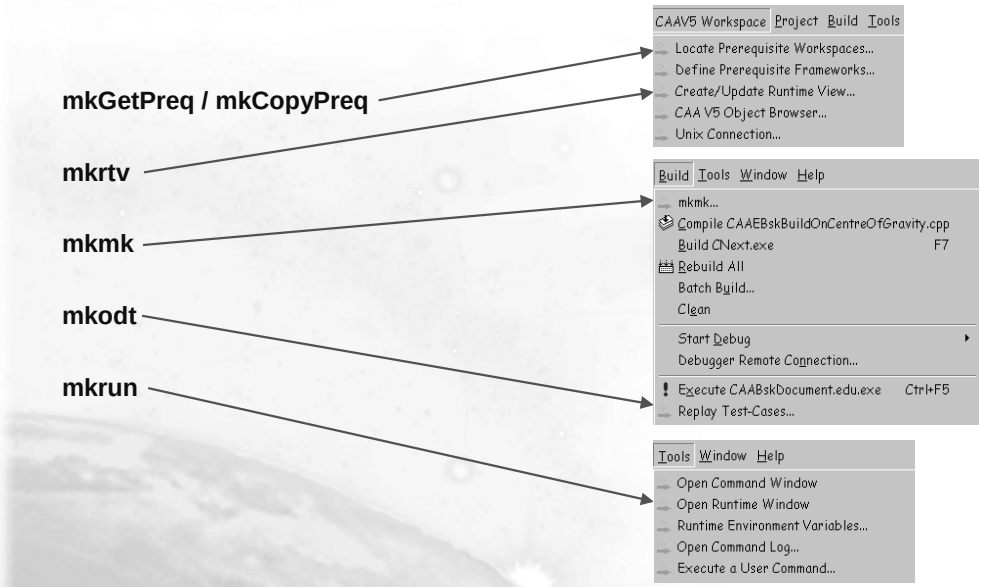


CAA V5 Object Browser

To know which component implements which interface

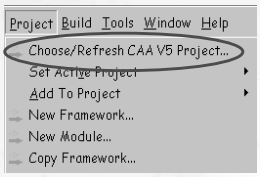


Mapping between commands and MSDev Add-Ins



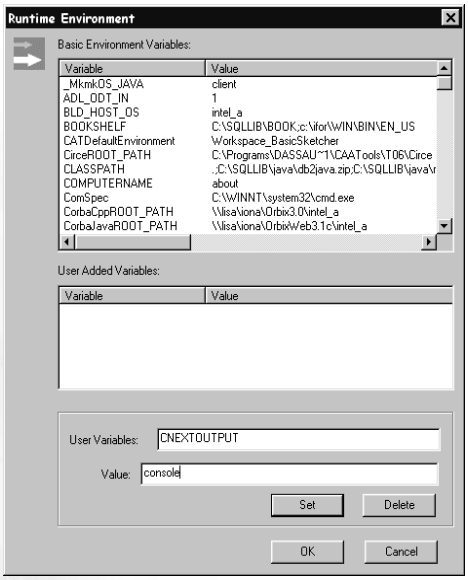
MSDev Add-Ins: Hints and Tips (1/3)

To be able to see any modification done directly in NT Explorer (copied, moved or deleted files) in MSDev,
Use command Project + Choose/Refresh CAA V5 Project…



MSDev Add-Ins: Hints and Tips (2/3)

To be able to see your trace statements, the environment variable CNEXTOUTPUT has to be set to « console » in Tools + Runtime Environment Variables



MSDev Add-Ins: Hints and Tips (3/3)

To rebuild a module and if you don't need the update option, use the keyboard shortcut F7.

To export a workspace (just the source code) get rid of all the intermediate files generated by mkmk:
Go to Tools + Open Command Window and key
mkRemoveDo -a

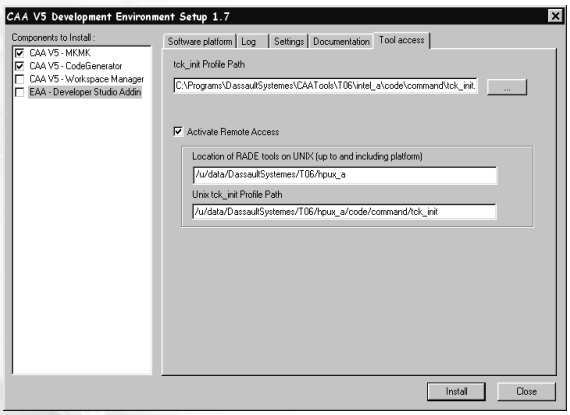
[Ctrl-Q] to swap between .h and .cpp files
[Ctrl-T] to open the .h file corresponding to the keyword under the cursor
[Ctrl-F1] for API documentation

Enable porting on UNIX from Visual C++

If you need to port your applications on UNIX, you should run again the CATVBTSetup executable:

C:\Programs\DassaultSystemes\T06\intel_a\code\bin

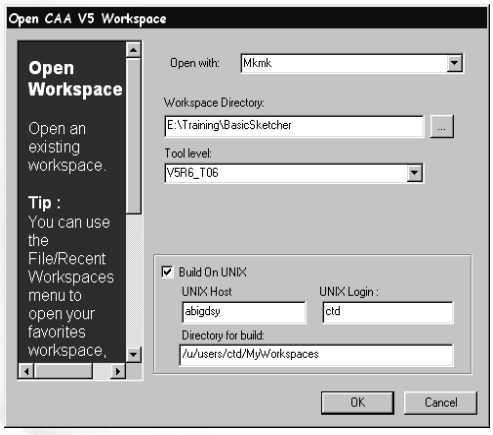
In the Tool access tab page, activate the remote access and inform where the CAA V5 Tools are installed on UNIX



Activate the Porting on UNIX

When opening a workspace, you can ask for building on UNIX by informing Visual C++ on which UNIX machine, with which user and in which directory the operations will be performed.

Later on, whenever a file is generated on NT, it is copied on UNIX



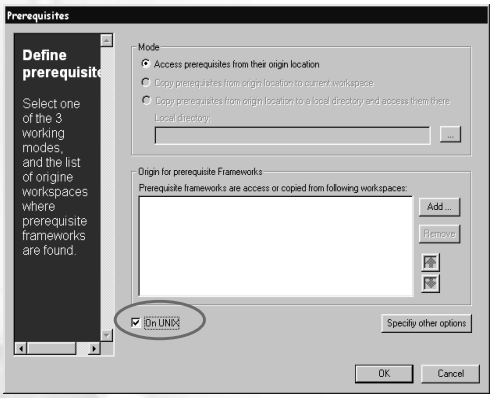
Porting on UNIX

From Visual C++, then

you can define the prerequisite workspaces

you can build

you can update the run time view



Other Tools used in the CAA V5 context

Workspace Manager

A Dassault Systèmes tool to manage the source code versioning and to organize and control software developments between development departments.

Rational Purify

A tool to detect any memory leak and to be used with mkodt

Rational Pure Coverage

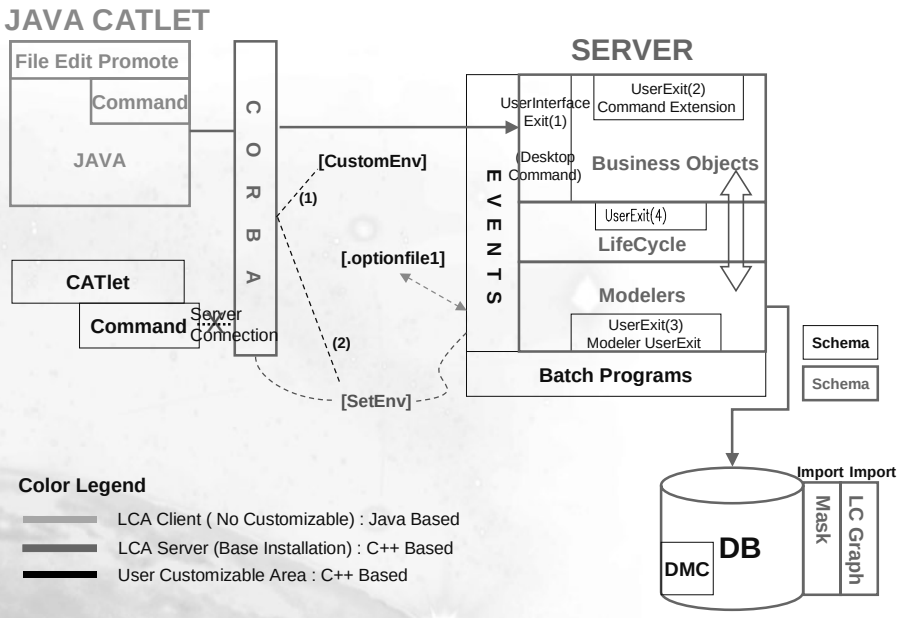
A tool to check the percentage of the code really tested and to be used with mkodt

Customization of ENOVIA LCA

You will see to find information's on CAA V5

- Customization of ENOVIA LCA
- Setting the environment for ENOVIA LCA
- Customizing ENOVIA LCA

Customization of ENOVIA LCA



Setting the environment for ENOVIA LCA

Launching of the client ENOVIA LCA on UNIX

under $RUN_TIME_VIEW/$OS/code/command Launch

./enoviastart –env ENOVIA_LCA.V5R9.B09.sh -d /CATEnv

where the option

-env : name of the environment shell scrip ENOVIA_LCA.V5R9.B09.sh which set the environment path

-d : locate the shell on the UNIX machine

Settings the path for the Run Time View on UNIX

under $RUN_TIME_VIEW/$OS/code/command Launch

./SetEnv –env ENOVIA_LCA.V5R9.B09.sh -d /CATEnv

where the option

-env : name of the environment shell scrip ENOVIA_LCA.V5R9.B09.sh which set the environment path

-d : locate the shell on the UNIX machine

Customizing of ENOVIA LCA

When you chose a profile on ENOVIA LCA, ENOVIA launch the file shell scrip CustomEnv under the home directory of the profile

In this file RADE set the Customization path

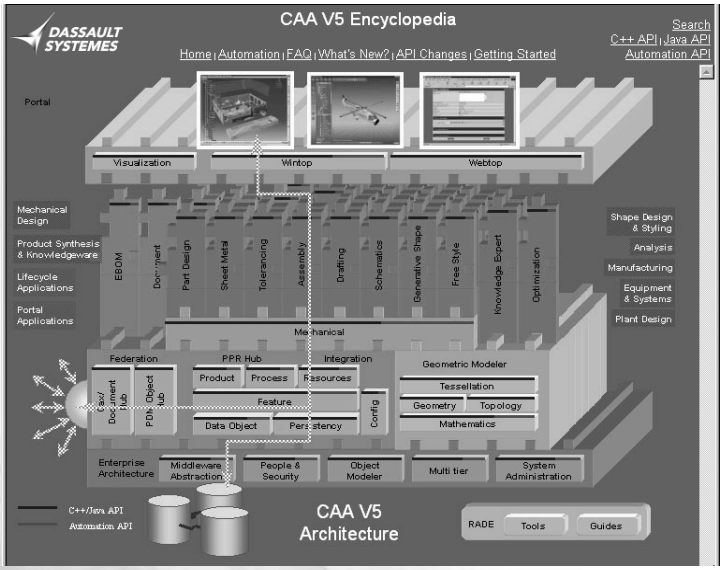
```
export MkmkOS_VAR=aix_a
export VPM_PLUGIN_OBJECTS_LIST="PluginList"

PATH=/home/vpm5adm/MSDEV/E__WSEVENTS/aix_a/code/bin:$PATH
export PATH
LIBPATH=/home/vpm5adm/MSDEV/E__WSEVENTS/aix_a/code/bin:$LIBPATH
export LIBPATH
LD_LIBRARY_PATH=/home/vpm5adm/MSDEV/E__WSEVENTS/aix_a/code/bin:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
SHLIB_PATH=/home/vpm5adm/MSDEV/E__WSEVENTS/aix_a/code/bin:$SHLIB_PATH
export SHLIB_PATH
CATDictionaryPath=/home/vpm5adm/MSDEV/E__WSEVENTS/aix_a/code/dictionary:$CATDictionaryPath
export CATDictionaryPath
CATMsgCatalogPath=/home/vpm5adm/MSDEV/E__WSEVENTS/aix_a/resources/msgcatalog:$CATMsgCatalogPath
export CATMsgCatalogPath
```

CAA V5 Encyclopedia and Programming Rules

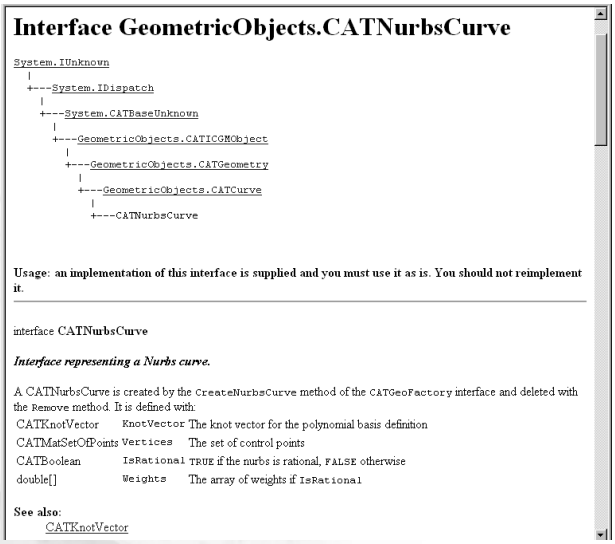
You will see to find information's on CAA V5

- CAA V5 Encyclopedia Home Page
- CAA V5 C++ Object Documentation
- CAA V5 Programmer's Guide
- CAA V5 Programming Rules

CAA V5 Encyclopedia Home Page



CAA V5 C++ Object Documentation



Interface GeometricObjects.CATNurbsCurve

```
System.IUnknown
  |
  +---System.IDispatch
    |
    +---System.CATBaseUnknown
      |
      +---GeometricObjects.CATICGMObject
        |
        +---GeometricObjects.CATGeometry
          |
          +---GeometricObjects.CATCurve
            |
            +---CATNurbsCurve
```

Usage: an implementation of this interface is supplied and you must use it as is. You should not reimplement it.

interface **CATNurbsCurve**

Interface representing a Nurbs curve.

A CATNurbsCurve is created by the `CreateNurbsCurve` method of the `CATGeoFactory` interface and deleted with the `Remove` method. It is defined with:

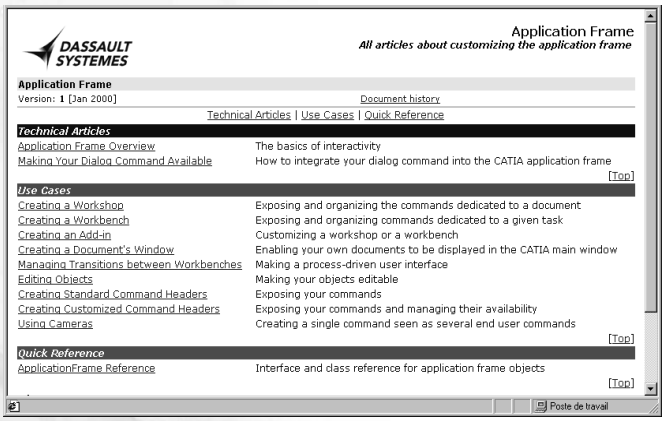
CATKnotVector	`KnotVector`	The knot vector for the polynomial basis definition
CATMathSetOfPoints	`Vertices`	The set of control points
CATBoolean	`IsRational`	`TRUE` if the nurbs is rational, `FALSE` otherwise
double[]	`Weights`	The array of weights if `IsRational`

See also:
CATKnotVector

CAA V5 Programmer's Guide (1/3)

All documentations about a domain

Technical Articles
Use Cases
Quick references



DASSAULT SYSTEMES

Application Frame
All articles about customizing the application frame

Application Frame
Version: **1** [Jan 2000] Document history
Technical Articles | Use Cases | Quick Reference

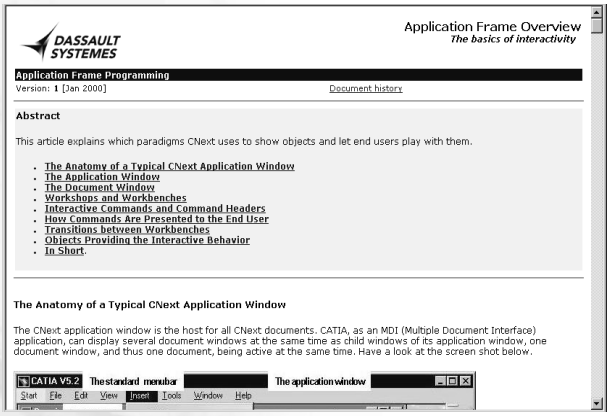
Technical Articles	
Application Frame Overview	The basics of interactivity
Making Your Dialog Command Available	How to integrate your dialog command into the CATIA application frame
	[Top]
Use Cases	
Creating a Workshop	Exposing and organizing the commands dedicated to a document
Creating a Workbench	Exposing and organizing commands dedicated to a given task
Creating an Add-in	Customizing a workshop or a workbench
Creating a Document's Window	Enabling your own documents to be displayed in the CATIA main window
Managing Transitions between Workbenches	Making a process-driven user interface
Editing Objects	Making your objects editable
Creating Standard Command Headers	Exposing your commands
Creating Customized Command Headers	Exposing your commands and managing their availability
Using Cameras	Creating a single command seen as several end user commands
	[Top]
Quick Reference	
ApplicationFrame Reference	Interface and class reference for application frame objects
	[Top]

Poste de travail

CAA V5 Programmer's Guide (2/3)

Technical articles

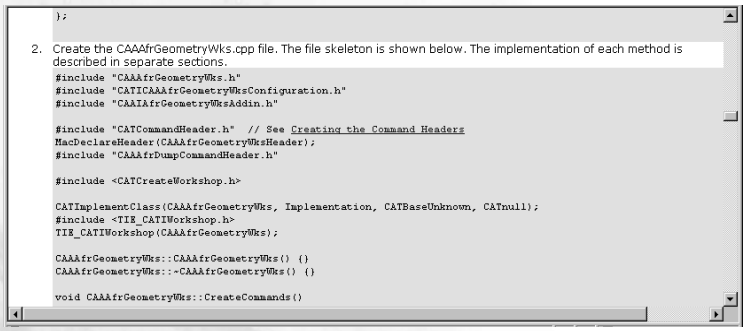
In depth paper
Less than 10 pages
Hyper linked



CAA V5 Programmer's Guide (3/3)

Use Cases

CAA V5 Code in Action
Step by Step
Each step detailed and commented
Delivered with fully operational source code
Made to be copied/pasted into customer code



CAA V5 Programming Rules

- ***Programming Rules***
  - ***Naming convention***
  - ***C++ coding rules***
  - ***Java coding rules***
  - ***Architecture rules***
  - ***User Interface look and feel***
- ***Available in the encyclopedia***

CAA V5 Naming Convention

- ***Naming conventions***
  - ***To avoid name collisions***
  - ***To make things clearer for its developers***
- ***Names are constituted by English names. Each one starts with an uppercase.***
- ***Three letters alias for product name.***
  - ***CAT / VPM / ENO / DNB reserved for Dassault Systèmes product lines***
- ***Three letters alias for each framework.***
  - ***CATMoldDesignFeature (framework)***
  - ***CATMldComponent.m (module)***
  - ***CATMldEjectorImpl.cpp/.h (class)***

CAA V5 C++ Programming Rules

- ***Prefer class forward declaration to #include***
- ***Avoid multiple inheritance***
- ***Use variable naming convention***

  Argument prefix: ***i for input, o for output***

  Variable prefix:

p	***pointer***
pp	***pointer on pointer***
sp	***smart pointer***
pi	***pointer on interface***
a	***array***

- ***Do not use friend class***

RADE Installation and Licensing

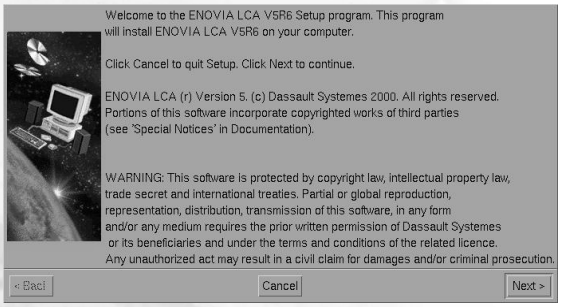
You will see the RADE Installation and Licensing

- **Softwares to download on UNIX**
- **Software to download on NT**

Softwares to download on UNIX (1/3)

RUN TIME

Install ENOVIA_LCA.$OSname
Choose on of the Configurations (RVR, ADR ….)

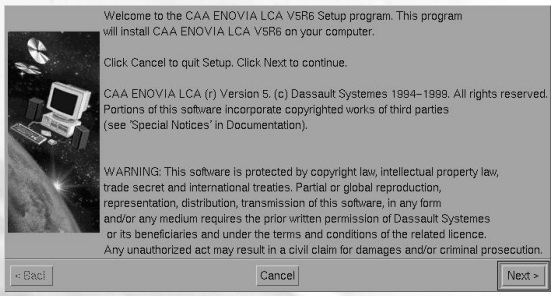


Softwares to download on UNIX (2/3)

BUILD TIME

Install CAA_ENOVIA_LCA.unix

This download contains :
COY : CAA ENCYCLOPEDIA
EAP : CAA ENOVIA LIFE CYCLE API Product



Softwares to download on UNIX (3/3)

TOOLS

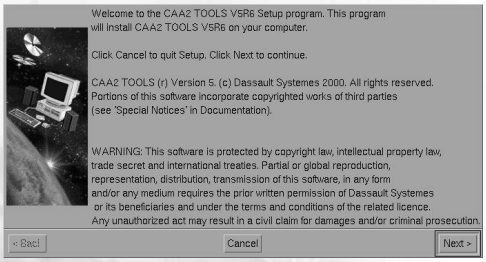
Install RADE.unix

This download contains the tools mkmk

You choose one the two Configurations

CDC CAA - C++ Development Configuration

LDC CAA - Legacy Data Integration Development Configuration



Software to download on NT (1/2)

TOOLS

Install RADE.intel_a

This download contains :

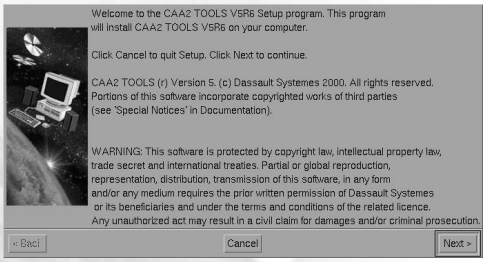
CAA2 Development Toolkit

CAA2 Data Motorization Customer

Legacy Rad

CAA2 Run Time Quality Control

CAA2 Visual Builder Toolkit



Software to download on NT (2/2)

Products

MsDev: 6.0

Rational Rose: 98.0 and next versions (with local license, « Rational Rose Modeler 2001 » is needed)

Rational Purify: V6.5

Rational Pure Coverage: V6.5NT version

MKS Toolkit: V6.1A (Korn Shell on NT)

SilverStream: Legacy RADE has no prerequisite on SilverStream.

Iona ORBIX: 3.0.

To Sum Up

In this course you have seen :

- **The CAA V5 directory tree structure**
- **The specific tools developed on top of the standard compilers to speed up the development**
- **The CAA V5 Encyclopedia and rules to help you to programming**
- **RADE Installation and Licensing**