Ministry of Agriculture & Land Reclamation
Agricultural Research Center
Central Lab for Agricultural Expert Systems

KSR Tool
Code Documentation

TR/CLAES/263/2003.4

Version 1.00

Eng.Mohammed El Helly

Eng Mohammed Yehia

Eng Mohammed Said

April, 2003

Introduction

Knowledge Share and Reuse (KSR) is an Expert system shell, which has been built using visual C++ to facilitate building the expert system at CLAES.

The tool has six editors; concept, rule, inference, table, function, database. In addition some functionality like, Create metaKB, Generate design document, Validate KB.

The first editor is the Concept editor, which is used to edit the concept, property and facet. This editor has five pages; the first page (concept page) is used to edit concepts only, the second page (property page) is used to edit the property of the concept, the third page (facet page) is used to edit the facet of each property, the fourth page (refine page) is used to update concept name and property name in all knowledge base if we discover mistake in it, the fifth page(image page) is used to associate image with concept or property or value.

The second editor is the Rule editor, is used to edit and debug the rule, it consists of two pages, rule page, and debug page, which is used to run the rule or cluster at development time.

The third editor is the Table editor, is used to edit knowledge base in the table format

The fourth editor is the function editor, is used to edit functions in the knowledge base.

The fifth editor is the database editor, is used to edit relation between the database and knowledge base.

The sixth editor is the inference editor, is used to edit different inference in the knowledge base.

The MetaKB is the functionality of generate knowledge base in the XML format that facilitate explanation.

The Generate Design Document is the facility to generate the knowledge in a readable html format.

The ValidateKB is the facility for checking any validation error in the knowledge base.

The purpose of this document is to describe each class used in the KSR, for technical usage, by the tool programmers.

The KSR tool consists of 48 classes; they are categorized by functionality into

- **MFC Framwork classes**

CKBEditorsApp	CKBEditorsDoc
CKBEditorsView	CMainFrame

This category include the following classes

- **User defined classes**

The user defined classes is divided into three sub category

  - **Data Structure Classes**

CConcept	CProperty	CValue	CValueList
CRule	CCluster	CInference	CFunction
CTable	CCell	CImageOb	CDBHandle
CKBDB	CTblKeys	FunParser	FunToken
CTreeItem	CXMLRPC	CXmlUtils	

  - **Interface Classes**

CCptPage	CPropPage	CFacetPage
CRulePage	CDebugPage	CRuleSheet
CFunSheet	CDesignDocument	CImage
CKBDBDlg	CMetaKBdlg	CNominalDlg
CTableDlg	CTableNameDlg	OperationVlaueDlg
AddDeleteDlg	CRefineKB	CInferenceDlg
CCptSheet	CFunDlg	CNumberDlg
CDateDlg		

  - **OCX Classes**

CMSFlexGrid	CWebBrowser2

we will describe each of them in terms of its purpose, property, and method. The interface class will be described in the MiniKSR document.

Before we describe each class we shall describe the new data types added to facilitate building complex data structure required to build knowledge base.

New Types

The following types are used in the KSR and have been build using MFC classes. To facilitate the handling of KSR collections classes

CCptList

This type is the collection of CConcept class. The input of the collection is the "concept" name. The output of the collection is pointer to the CConcept class. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString,CConcept> CCptList*

CPropList

This type is collection of CProperty class. The input of the collection is the "concept-property" name. The output of the collection is pointer to the CProperty class. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString,CProperty> CPropList;*

CLegalValueList

This type is collection of CString class. The input of the collection is the "CString" represent an English name of the legal values. The output of the collection is CString class represents the Arabic name of that legal. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString, CString >CLegalValueList

CRuleList

This type is collection of CRule class. The input of the collection is the "Rule ID" name. The output of the collection is pointer to the CRule class. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString, CRule> CRuleList*

CClusterList

This type is collection of CCluster class. The input of the collection is the "Cluster ID" name. The output of the collection is pointer to the CCluster class. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString, CCluster> CClusterList*

CInferenceList

This type is collection of CInference class. The input of the collection is the "Inference ID" name. The output of the collection is pointer to the CInference class. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString, CInference>CInferenceList*

CWMStructure

This type is collection of CValueList class. The input of the collection is the "concept-property" name. The output of the collection is pointer to the CValueList class. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString, CValueList> CWMStructure*

CKBDBList

This type is collection of CKBDB class. The input of the collection is the "ConceptPropertyTableField" names. The output of the collection is pointer to the CKBDB class. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString,CKBDB> CKBDBList;*

CTblKeysList

This type is collection of CTblKeys class. The input of the collection is the "Table" name. The output of the collection is pointer to the CTblKeys class. The Specification of the that collection is as follow

CTypedPtrMap<CMapStringToOb,CString,CTblKeys> CTblKeysList;*

1- Class Name CConcept

1-1 Description:

> *This class is used to hold the information about the concept which is used to build knowledge base, like leave, stem, plant, etc…*

1-2 Properties:

Name	Type	Usage
NameL;	CString	English Name
SupperNID	long	Parent Identification
nID	long	Unique Concept Identification
NameA	CString	Arabic Name
DescriptionA	CString	Arabic Description
DescriptionL	CString	English Description
m_PropList	CPropList	Properties of the Concept

This class has only one method

1-3 Method Name "Serialize"

This method is used to save and load the properties of that concept on hard disk.

Input	Type	Meaning
ar	CArchive	Archive class (high level file object)

2- Class Name CProperty

2-1 Description:

> *This class is used to define the properties (facets) of the concept*

2-2 Properties:

Name	Type	Meaning
NameL	CString	English Property Name
nID	long	Unique Property Identification
NameA	CString	Arabic Property Name
PromptA	CString	Arabic Prompt
PromptL	CString	English Prompt
ControlID	int	Not used
Type	CString	Property Type (integer, real,etc.)
SourceOfValue	CString	From which the inference get the property value during inference process
Default	CString	Default value of the property is used during inference process
UL	float	Upper limit of the property in case of numeric property
LL	float	Lower limit of the property in case of numeric property
m_LegalValueList	CLegalValueList	Possible legal values of the property
m_SearchOrderList	CStringList	Ordered list to guide the inference how to get the value of that property during inference process (Not implemented)
Necessary	BOOL	Indicate the importance of that property during inference process

This class has only one method

2-3 Method Name "Serialize"

This method is used to save and load the properties of that property class on hard disk.

Input	Type	Usage
ar	CArchive	Archive class (high level file object)

3- Class Name CRule

3-1 Description:

> *This class is used to store the rule structure as a part of the knowledge base*

3-2 Properties:

Name	Type	Meaning
RuleID	CString	Rule Identification
Priority	int	Rule priority in case of firing rule by its priority
Condition	CString	String represent rule condition
Action	CString	String represent rule Action
pView	CKBEditorsView*	Pointer to the view class
m_Doc	CKBEditorsDoc*	Pointer to the document class

3-3 Methods

This class has 31 methods

3-3-1 Serialize

This method is used to save and load the rule class on hard disk.

Input	Type	Usage
ar	CArchive	Archive class (high level file object)

3-3-2 CRule

Constructor of the rule class.

Input	Type	Usage
pDoc	CKBEditorsDoc*	Used to init. The pointer to the document when create an instance of the rule to make it possible to access on the document class

3-3-3 GetValue

Retrieve from working memory the value of the specific key.

Input	Type	Usage
str	CString	String represent a key in working memory structure, this method is used in database to retrieve the value of the key, this key is not a concept name
m_WM	CWMStructure*	Pointer to working memory to be searched

Return Type	Description
CString	The value of the key, if the key is not exist an empty string will be returned

3-3-4 GetType

Retrieve the type of the property from the stored knowledge base.

Input	Type	Usage
str	CString	String represent a key in concept list structure, the format of that *str* is "Cpt-Prop"
pCptList	CCptList*	Pointer to the concept list to be searched

Return Type	Description
CString	String represents the type of the property. It is one of the following values *"integer", "real", "nominal", "Multi-Value", "boolean", "date", "string"*

3-3-5 DecomposeCptProp

This method decomposes the concept property string into a concept string and property string.

Input	Type	Usage
str	CString	String in the format "Cpt-Prop"(input)
ssCpt	CString &	Copcept string name (output)
ssProp	CString &	Property string name (output)

3-3-6 CEvalExp

This method evaluate an expression (Mathematical or Boolean)and return double value.

Input	Type	Usage
str	CString	Expression in string format

Return Type	Description
double	The value of calculated expressions

3-3-7 CEval

This method evaluate an expression (Mathematical or Boolean)and return Boolean value.

Input	Type	Usage
str	CString	Expression in string format

Return Type	Description
Bool	The value of calculated expressions

3-3-8 GetValue

This method search in working memory and get the value of the property, if there is no value in the working memory the default value of that property is retrieved.

Input	Type	Usage
Cpt	CString	String represent a concept name
Prop	CString	String represent a property name
List	CStringList &	List of string values
m_WM	CWMStructure *	Working memory to be searched
pCptList	CCptList*	Pointer to concept list
Return Type	**Description**	
CStringList	List of string value of the concept property	

3-3-9 GetValue

This method search in working memory and get the value of the property, if there is no value in the working memory, empty string will be returned

Input	Type	Usage
Cpt	CString	String represent a concept name
Prop	CString	String represent a property name
M_WM	CWMStructure *	Working memory to be searched

Return Type	Description
Cstring	The value of the property, or an empty string if that property does not exist

3-3-10 GetDefaultValue

This method return the default value of the property, otherwise it return an empty string

Input	Type	Usage
strCpt	CString	String represent a concept name
strProp	CString	String represent a property name
pCptList	CCptList*	Concept list to be searched

Return Type	Description
Cstring	The default value of the property, or an empty string if that property does not have a default

3-3-11 StrCptPropIn

This method search on str in str2, and return true if str existed in str2, otherwise it return false. The formate of str is"$Cpt.Prop" and the format of str2 is “"$Cpt.Prop"$Cpt.Prop……”. This method is called by AskingCheck method during inference process to ask user about the unknown property

Input	Type	Usage
str	CString	String represent a concept property in format “$cpt.prop"name
Str2	CString	String represent a concept property in format “$cpt1.prop1$cpt2.prop2…..”

Return Type	Description
Bool	True if str is in str2 otherwise return false

3-3-12 AskUser

This method displays a dialogue to get the value of the properties (existed in input parameter str in format “$cpt.prop……”) from the user and set this value to working memory. If this property already has a value in working memory then the dialogue will not appear, and the string containing a values of those properties will be returned

Input	Type	Usage
str	CString	String represent a concept property in format "$cpt.prop……"
m_WM	CWMStructure *	Pointer to working memory
pCptList	CCptList*	Pointer to concept list

Return Type	Description
CString	The value of the properties if they existed in working memory

3-3-13 AskingCheck

This is an intelligent method used to check the condition of the rule if it is contains unknown properties and those properties required to inference process or not.

Input	Type	Usage
strEval	CString	String represent a concept property in format "$cpt.prop……"
m_WM	CWMStructure *	Pointer to working memory
pCptList	CCptList*	Pointer to concept list

Return Type	Description
CString	Expression in string format to be evaluated by inference

3-3-14 SetToWM

This method set the value of the property into working memory under the key consists of "strCpt-strProp".

Input	Type	Usage
strCpt	CString	The name of the concept
strProp	CString	The name of the property
strVal	CString	The name of the value
m_WM	CWMStructure *	Pointer to working memory
pCptList	CCptList*	Pointer to concept list

3-3-15 Run

This method fire the rule, and return true or false based on the condition of that rule.

Input	Type	Usage
m_WM	CWMStructure *	Pointer to working memory
pCptList	CCptList*	Pointer to concept list

Return Type	Description
Bool	True if the condition of the rule is succeeded otherwise it return a false

3-3-16 GetCondition

This method return string condition of the rule

Return Type	Description
Cstring	String represent the condition of the rule

3-3-17 GetAction

This method return string action of the rule

Return Type	Description
Cstring	String represent the action of the rule

3-3-18 SetCondition

This method set the condition to the rule object

Input	Type	Usage
Cond	CString	String represent the condition of the rule

3-3-19 SetAction

This method set the Action to the rule object

Input	Type	Usage
Act	CString	String represent the action of the rule

3-3-20 StrToDate

This method Convert a string to date

Input	Type	Usage
str	CString	String represent the action of the rule
Expr	int	Number of days added to the date
Return Type	**Description**	
CTime	Date object	

3-3-21 GetType

Retrieve the type of the property from the stored knowledge base.

Input	Type	Usage
strCpt	CString	Concept name
strProp	CString	Property name
pCptList	CCptList*	Pointer to the concept list to be searched

Return Type	Description
CString	String represents the type of the property. It is one of the following values *"integer", "real", "nominal", "Multi-Value", "boolean", "date", "string"*

3-3-22 ReplaceNow

Replace now string in the input string by the current date in that string.

Input	Type	Usage
str	CString	String contains a now function

Return Type	Description
CString	String without now, the now keyword is replaced with the current date in string format

3-3-23 AddActionToWM

Parse the action and executes every command of it.

Input	Type	Usage
strAction	CString	An action string
m_WM	CWMStructure *	Pointer to working memory
pCptList	CCptList *	Pointer to concept list

3-3-24 ReplaceAbsDate

Convert a string date to a string days.

Input	Type	Usage
str	CString	String date in format #00/00/00#

Return Type	Description
CString	Number of days in the input

3-3-25 EvalDateFun

Evaluate a a function of date expressions like month year and day and return the value in string format.

Input	Type	Usage
str	CString	Date expression contains month or year or day in a string format
Return Type	**Description**	
CString	Result of evaluation in a string format	

3-3-26 EvalDate

Evaluate a date expressions and return the value in string format.

Input	Type	Usage
str	CString	Date expression in a string format

Return Type	Description
CString	Result of evaluation in a string format

3-3-27 Substitute

This method parses the condition and replaces each concept property with its value from working memory, and return substituted condition in a string format.

Input	Type	Usage
str	CString	Condition of the rule
m_WM	CWMStructure *	Pointer to working memory
pCptList	CCptList*	Pointer to the concept list

Return Type	Description
CString	Substituted condition in a string format.

3-3-28 EvalFunction

This method evaluates an expression in a string format, which contains concept property or more and returns a value in a string format

Input	Type	Usage
strExpr	CString	Expression contains concept property or more
m_WM	CWMStructure *	Pointer to working memory

Return Type	Description
CString	A value of the evaluated expression.

3-3-29 RemoveSpace

Remove a white space from the condition of the rule before evaluate them

Input	Type	Usage
strCond	CString	Condition of the rule

Return Type	Description
CString	Condition without space.

3-3-30 ValidateDateFun

Parse a string date to check if it contains a date function in a valid format or not

Input	Type	Usage
Str	CString	String date

3-3-31 ValidateDate

Parse a string date to check if it in a valid format or not

Input	Type	Usage
str	CString	String date
Return Type	**Description**	
CString	True if it is valid otherwise not	

4- Class Name CCluster

4-1 Description:

> *This class is used to store the collection of rules*

4-2 Properties:

Name	Type	Meaning
ClusterID	CString	Cluster Identification
Input	CObList	List of input of that cluster

Output	CObList	List of output of that cluster
RuleList	CRuleList	Rule List of that cluster
Rule	CRule *	Pointer to the CRule class

4-3 Methods

This class has 5 methods

4-3-1 Serialize

This method is used to save and load the Cluster class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

4-3-2 RunByPriority

This method sorts the rules in the cluster according to its priority and then run those sorted rules.

Input	Type	Usage
m_WM	CWMStructure *	Pointer to working memory structure
pCptList	CCptList*	Pointer to the concept list

4-3-3 Run

This method runs each rule in the cluster regardless its priority

Input	Type	Usage
m_WM	CWMStructure *	Pointer to working memory structure
pCptList	CCptList*	Pointer to the concept list

4-3-4 RunRule

This method lookup on the rule ID stored in parameter strRule and then call rule method to run this rule

Input	Type	Usage
m_WM	CWMStructure *	Pointer to working memory structure
strRule	CString	String represent rule ID
pCptList	CCptList*	Pointer to the concept list

4-3-5 RunRule

This call rule method to run this rule, and return true or false.

Input	Type	Usage
m_WM	CWMStructure *	Pointer to working memory structure
pCptList	CCptList*	Pointer to the concept list

Return Type	Description
BOOL	True if the rule succeeded, otherwise false.

5- Class Name CInference

5-1 Description:

> *This class is used to store the collection of Cluster*

5-2 Properties:

Name	Type	Meaning
InferenceID	CString	Inference Identification
StaticRoleList	CStringList	Cluster List of that inference

5-3 Methods

This class has one method

5-3-1 Serialize

This method is used to save and load the CInference class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

6- Class Name CValue

6-1 Description:

> *This class is used to store the one value to working memory structure*

6-2 Properties:

Name	Type	Meaning
V	CString	The name of the value

6-3 Methods

This class has one method

6-3-1 Serialize

This method is used to save and load the CValue class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

7- Class Name CValueList

7-1 Description:

> *This class is used to store the List of values to working memory structure*

7-2 Properties:

Name	Type	Meaning
m_CValueList	CObArray	Array of objects contains the values to be saved

7-3 Methods

This class has one method

7-3-1 Serialize

This method is used to save and load the CValueList class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

8- Class Name CCptSheet

8-1 Description:

> *This class is used to display the concept editor on the screen,The concept editor consists of group of pages. Each page have a class to maintain it, we will describe each class individually*

8-2 Properties:

Name	Type	Meaning
CptDlgExist	BOOL	Flag used to indicate the appearance of the concept editor on the screen, the tool permits one instance only of the concept editor to appear on the screen
m_CCptPage	CCptPage	Instance to the concept page
m_CPropPage	CPropPage	Instance of the property page
m_CFacetPage	CFacetPage	Instance to the facet page
m_RefineKBPage	CRefineKB	Instance to the refinement page
m_ImagePage	CImage	Instance to the image page
m_Doc	CKBEditorsDoc *	Pointer to the document class

8-3 Methods

This class has five methods

8-3-1 CCptSheet

Constructor of the class , init the caption, default page, parent window of the sheet

Input	Type	Usage
nIDCaption	UINT	Id of caption if the property sheet
pParentWnd	CWnd*	Pointer to the parent class of property sheet default NULL
iSelectPage	UINT	The default page to appear default zero

8-3-2 CCptSheet

Overloaded constructor of the class , init the caption, default page, parent window of the sheet

Input	Type	Usage
pszCaption	LPCTSTR	String caption if the property sheet
pParentWnd	CWnd*	Pointer to the parent window of property sheet default NULL
iSelectPage	UINT	The default page to appear default zero

8-3-3 SeeDoc

This method used to initiate the pointer to the document class when creating concept sheet

Input	Type	Usage
Doc	CKBEditorsDoc*	Pointer to the document class

8-3-4 OnInitDialog

The framework of MFC classes calls this method; we use it to pass the pointer document to each concept sheet page

Return Type	Description
BOOL	True if the sheet appeared successfully, otherwise false.

8-3-5 DestroyWindow

The framework of MFC classes calls this method; we use it to init the concept sheet flag

Return Type	Description
BOOL	True if the sheet destroyed successfully, otherwise false.

9- Class Name CRuleSheet

9-1 Description:

> *This class is used to display the Rule editor on the screen; the Rule editor consists of two pages. Each page have a class to maintain it, we will describe each class individually*

9-2 Properties:

Name	Type	Meaning
RuleDlgExist	BOOL	Flag used to indicate the appearance of the Rule editor on the screen, the tool permits one instance only of the rule editor to appear on the screen
m_RulePage;	CRulePage	Instance to the Rule page
m_DebugPage	CDebugPage	Instance of the debug page
m_Doc	CKBEditorsDoc *	Pointer to the document class

9-3 Methods

This class has five methods

9-3-1 CRuleSheet

Constructor of the class , init the caption, default page, parent window of the sheet

Input	Type	Usage
nIDCaption	UINT	Id of caption if the property sheet
pParentWnd	CWnd*	Pointer to the parent class of property sheet default NULL
iSelectPage	UINT	The default page to appear default zero

9-3-2 CRuleSheet

Overloaded constructor of the class, init the caption, default page, parent window of the sheet

Input	Type	Usage
pszCaption	LPCTSTR	String caption if the property sheet
pParentWnd	CWnd*	Pointer to the parent window of property sheet default NULL
iSelectPage	UINT	The default page to appear default zero

9-3-3 SeeDoc

This method used to initiate the pointer to the document class when creating Rule sheet

Input	Type	Usage
Doc	CKBEditorsDoc*	Pointer to the document class

9-3-4 OnInitDialog

The framework of MFC classes calls this method; we use it to pass the pointer document to each rule sheet page

Return Type	Description
BOOL	True if the sheet appeared successfully, otherwise false.

9-3-5 DestroyWindow

The framework of MFC classes calls this method; we use it to init the rule sheet flag

Return Type	Description
BOOL	True if the sheet destroyed successfully, otherwise false.

10- Class Name CDateDlg

10-1 Description:

> *This class is used to display the Dialogue to get the value of any property of type date required during reasoning process*

10-2 Properties:

Name	Type	Meaning
m_Date	CTime	Member variable associated with date control used to hold the value the property date
m_Prompt	CString	Member variable associated with textbox control used to hold the prompt of the property
Prop	CProperty *	Pointer to the CProperty object
Val	CString	Store the acquired date into the string format

10-3 Methods

This class has five methods

10-3-1CDateDlg

Constructor of the dialogue

Input	Type	Usage
pParent	CWnd*	Pointer to the parent window (default NULL)

10-3-2DoDataExchange

This method created by the framework of MFC weathered, is used to make an association between the visual interface appear to the user and member variables

Input	Type	Usage
pDX	CDataExchange*	Pointer to the data exchange class

10-3-3OnInitDialog

This event handler is called when the dialogue appear on the screen, used to init the member variables of the dialogue class

Return Type	Description
BOOL	True if the dialogue appear successfully, otherwise false.

10-3-4SeeData

This method is used to init the CProperty pointer that is used by dialogue to retrieve the prompt of that property.

Input	Type	Usage
inProp	CProperty *	Pointer to the CProperty class

10-3-5OnOK

This event handler is used to close the dialogue and store the value gets from the user in the working memory

11- Class Name CNominalDlg

11-1 Description:

> *This class is used to display the Dialogue to get the value of any property of type nominal required during reasoning process*

11-2 Properties:

Name	Type	Meaning
m_LegalValue	CListBox	Member variable associated with List box control used to hold the legal values of the property nominal
m_Prompt	CString	Member variable associated with textbox control used to hold the prompt of the property
Prop	CProperty *	Pointer to the CProperty object
Val	CString	Store the acquired value into the string format

11-3 Methods

This class has five methods

11-3-1 CNominalDlg

Constructor of the dialogue

Input	Type	Usage
pParent	CWnd*	Pointer to the parent window (default NULL)

11-3-2 DoDataExchange

This method created by the framework of MFC weathered, is used to make an association between the visual interface appear to the user and member variables

Input	Type	Usage
pDX	CDataExchange*	Pointer to the data exchange class

11-3-3 OnInitDialog

This event handler is called when the dialogue appear on the screen, used to init the member variables of the dialogue class

Return Type	Description
BOOL	True if the dialogue appear successfully, otherwise false.

11-3-4 SeeData

This method is used to init the CProperty pointer that is used by dialogue to retrieve the prompt of that property.

Input	Type	Usage
inProp	CProperty *	Pointer to the CProperty class

11-3-5 OnOK

This event handler is used to close the dialogue and store the value gets from the user in the working memory

12- Class Name CNumberDlg

12-1 Description:

> *This class is used to display the Dialogue to get the value of any property of type number required during reasoning process*

12-2 Properties:

Name	Type	Meaning
m_Val	CString	Member variable associated with Textbox control used to hold the value of the property number
m_Prompt	CString	Member variable associated with textbox control used to hold the prompt of the property
Prop	CProperty *	Pointer to the CProperty object
Val	CString	Store the acquired value into the string format

12-3 Methods

This class has five methods

12-3-1CNominalDlg

Constructor of the dialogue

Input	Type	Usage
pParent	CWnd*	Pointer to the parent window (default NULL)

12-3-2DoDataExchange

This method created by the framework of MFC weathered, is used to make an association between the visual interface appear to the user and member variables

Input	Type	Usage
pDX	CDataExchange*	Pointer to the data exchange class

12-3-3OnInitDialog

This event handler is called when the dialogue appear on the screen, used to init the member variables of the dialogue class

Return Type	Description
BOOL	True if the dialogue appear successfully, otherwise false.

12-3-4SeeData

This method is used to init the CProperty pointer that is used by dialogue to retrieve the prompt of that property.

Input	Type	Usage
inProp	CProperty *	Pointer to the CProperty class

12-3-5OnOK

This event handler is used to close the dialogue and store the value gets from the user in the working memory

13- Class Name CKBDB

13-1 Description:

> *The responsibility of CKBDB class is to associate specific concept-property with specific table-field if their type matched*

13-2 Properties:

Name	Type	Meaning
Cpt	CString	String represent concept name
Prop	CString	String represent property name
Table	CString	String represent table name
Field	CString	String represent field name

13-3 Methods

This class has three methods

13-3-1Serialize

This method is used to save and load the CKBDB class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

13-3-2CKBDB

Constructor of the class used to init the property of the instance of CKBDB.

Name	Type	Meaning
strCpt	CString	String represent concept name
strProp	CString	String represent property name
strTable	CString	String represent table name
strField	CString	String represent field name

13-3-3GetCptPropTblFld

This method returns the values of the protected property of the class.

Name	Type	Meaning
C	CString&	String represent concept name
P	CString&	String represent property name
T	CString&	String represent table name
F	CString&	String represent field name

14- Class Name CTblKeys

14-1 Description:

> *The responsibility of CTblKeys class is to hold the list of keys of specific table and also list of their types*

14-2 Properties:

Name	Type	Meaning
TblKeys	CStringArray	String array hold a list of table keys name
TblKeysType	CStringArray	String array hold a list of table keys type

14-3 Methods

This class has three methods

14-3-1Serialize

This method is used to save and load the CTblKeys class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

14-3-2CTblKeys

Constructor of the class used to init the property of the instance of CTblKeys.

Name	Type	Meaning
Keys	CstringArray&	String array represent list of keys name
KeysType	CstringArray&	String array represent list of keys type

14-3-3GetTblKeys

This method returns the values of the private property of the class.

Name	Type	Meaning
Keys	CstringArray&	String array represent list of keys name
KeysType	CstringArray&	String array represent list of keys type

14-3-4RemoveAllKeys

This method removes all keys in the instance.

15- Class Name CDBHandle

15-1 Description:

> *The responsibility of CDBHandle class is to provide all operations required to access the database*

15-2 Properties:

Name	Type	Meaning
m_pCatalog	_CatalogPtr	Pointer to the database catalog
m_pTable;	TablesPtr	Pointer to the database table
pConnection	ADODB::_ConnectionPtr	Pointer to the database connection
pRsTable	ADODB::_RecordsetPtr	Pointer to the recordset

TablesName	CStringArray	List hold a tables name in the database
FieldsName	CStringArray	List hold the fields name of specific table
FieldsType	CUIntArray	List hold the type for each field in the specific table

15-3 Methods

This class has eleven methods

15-3-1CDBHandle

Constructor of the class, used to init the catalog, table pointer to NULL

15-3-2~CDBHandle

Destructor of the class used to remove all elements of the lists of the instance

15-3-3OpenDBCatalog

This method open the database catalog from the database name stored in the input parameter.

Name	**Type**	**Meaning**
DBName	CString	String represent database name

15-3-4GetTables

This method fills the input array with the tables name in the opened database. This method should be called after calling OpenDBCatalog method

Name	**Type**	**Meaning**
TablesArray	CStringArray&	String array to be filled with tables name

15-3-5GetFields

This method fills the input array with the field's name of input table name of the opened database. This method should be called after calling GetTables method

Name	**Type**	**Meaning**
TableName	CString	String represent the table name
FieldsArray	CStringArray&	String array to be filled with fields name of specific table

15-3-6GetTableIndex

Return with the index of the input table name from the TablesName array

Name	Type	Meaning
TableName	CString	String represent the table name

Return Type	Description
int	Integer represent the table index in the TablesName array

15-3-7GetFieldIndex

Return with the index of the input field name from the FieldsName array

Name	Type	Meaning
FieldName	CString	String represent the field name

Return Type	Description
int	Integer represent the table index in the FieldsName array

15-3-8GetFieldType

Return with the type of the input field name

Name	Type	Meaning
FieldName	CString	String represent the field name

Return Type	Description
CString	String represents the type of the input field name, one of the following values. *integer, boolean, string, date, real, Unknown*

15-3-9GetTableKeys

This method fills the input array with keys name of the specific table

Name	Type	Meaning
TableName	CString	String represent the table name
KeysArray	CStringArray&	String array to be filled with keys name of specific table

15-3-10 OpenDBTable

This method open the specific table and display each raw of its data to the user

Name	Type	Meaning
DBName	CString	Database name
Table	CString	Table name

15-3-11 GetDBValue

This method perform a query on the specific table to get the value of specific field

Name	Type	Meaning
DBName	CString	Database name
Table	CString	Table name
Field	CString	Field name
KeysName	CStringArray&	Array contains the keys name of that table
KeysValue	CStringArray&,	Array contains the keys value of that table
KeysType	CstringArray &	Array contains the keys type of that table

Return Type	Description
CString	String represent the value of the specific field

16- Class Name CMSFlexGrid

16-1 Description:

> *The Microsoft FlexGrid (**MSFlexGrid**) control displays and operates on tabular data. It allows complete flexibility to sort, merge, and format tables containing strings and pictures. This class is added when we add **MSFlexGrid.OCX** in any dialogue. This class has many methods we use just few of them and we will describe those methods only*

16-2 Methods

The seven methods used are described

16-2-1GetRows

Return with a number of rows in grid

16-2-2Refresh

Forces a complete repaint of a grid control

16-2-3RemoveItem

Removes a row from an MSHFlexGrid at run time

16-2-4SetTextMatrix

Set a string into specific cell

16-2-5GetRow

Return an index of a selected row

16-2-6SetRows

Init or update the total number of rows in the grid

16-2-7GetTextMatrix

Set the string at a specific cell

17- Class Name CWebBrowser2

17-1 Description:

> *The WebBrowser control adds browsing, document viewing, and data downloading capabilities to the application. This class is added when we add **shdocvw.dll** in any dialogue. This class has many methods we use just one method and we will describe this method only*

17-2 Methods

The only method used are described

17-2-1Navigate

Display a string in an HTML format in the browser control

18- Class Name CKBEditorsApp

18-1 Description:

> *It is the main application class in MFC encapsulates the initialization, running, and termination of an application for Windows. It is created automatically by application Wizard. An application built on the framework must have one (and only one) object of a class derived from CWinApp This object is constructed before windows are created. For more details return to the MFC documentation in MSDM*

18-2 Methods

The only method used are described

18-2-1 InitInstance

The standard InitInstance implementation created by AppWizard performs the following tasks:

- As its central action, creates the document templates that, in turn, create documents, views, and frame windows.
- Loads standard file options from an .INI file or the Windows registry, including the names of the most recently used files.
- Registers one or more document templates.
- Processes the command line to open a document specified on the command line or to open a new, empty document

19- Class Name CMainFrame

19-1 Description:

> *The MFC framework uses frame windows to contain views. The two components—frame and contents—are represented and managed by two different classes in MFC. A frame-window class manages the frame, and a view class manages the contents. The view window is a child of the frame window, for more details return to the MFC documentation in MSDN*

19-2 Methods

The created methods is:

19-2-1 OnCreate

This member function is called by the framework to allow the application to handle a Windows message. The parameters passed to your function reflect the parameters received by the framework when the message was received

19-2-2 PreCreateWindow

This member function is created automatically by application wizard and is called by the framework before the creation of the Windows; the update code is added to this method to adjust the position of the created window.

20- Class Name CKBEditorsView

20-1 Description:

> *AppWizard gives us a head start on the program development by creating skeletal document and view classes. We can then use Class Wizard to map commands and messages to these and to write their member functions, for more details return to the MFC documentation in MSDN.*
>
> *This class is responsible for display the main KSR toolbar*

20-2 Methods

The class has eleven methods, are described below, in addition eleven method created by framework and not used:

17-2-1OnNewKB

This method create an empty KB

17-2-2OnConcept

This method displays the concept editor on the screen.

17-2-3OnRule

This method displays the Rule editor on the screen.

17-2-4OnInference

This method displays the inference editor on the screen.

17-2-5OnTableEdit

This method displays the table editor on the screen.

17-2-6OnService

This method displays the database editor on the screen.

17-2-7OnCreateMetaKB

This method displays the Meta KB dialogue on the screen.

17-2-8OnGenerateDoc

This method displays the documentation dialogue on the screen.

17-2-9OnValidation

This method creates the validation report and displays it on the screen.

17-2-10 ValidateCondition

This method check the validity of condition of specific rule, and return with the string represents the error, if it is exist.

Name	Type	Meaning
strCond	CString	The condition of the rule
Out	CString&	Returning string represent the error

Return Type	Description
Bool	True if condition syntax is correct, otherwise false

17-2-11 ValidateAction

This method check the validity of action of specific rule, and return with the string represents the error, if it is exist.

Name	Type	Meaning
strAction	CString	The action of the rule
Out	CString&	Returning string represent the error

Return Type	Description
Bool	True if Action syntax is correct, otherwise false

21- Class Name CKBEditorsDoc

21-1 Description:

> *This class is created by application wizard and we added to it additional properties and method to support storing and retrieving the knowledge base structure from and to hard drive*

21-2 Properties:

Name	Type	Meaning
AppName	CString	Keep track the name of the knowledge base
AppDir	CString	Keep track the directory of the knowledge base
WorkingMemoryFile	CString	String represent name of working memory file
Cluster	CCluster *	Pointer to the CCluster

Cpt	CConcept *	Pointer to the CConcept
Rule	CRule *	Pointer to the CRule
Prop	CProperty *	Pointer to the CProperty
m_ImageListOb	CImageListOb	List of image objects
m_TempCptList	CCptList	Temporary concept list
m_CptListA	CCptList	Arabic concept list
m_MapIDToCpt	CCptIDList	List of Concept id’s
MaxConceptID	Long	Keep track the number of concepts in the knowledge base
MaxRuleID	Long	Keep track the number of rules in the knowledge base
Language	BOOL	To keep track the current language of the application English or Arabic language
m_ClusterList	CClusterList	List of clusters
m_InferenceList	CInferenceList	List of inferences
m_WM	CWMStructure*	Pointer to working memory structuer
m_TableList	CTableList	List of database tables associated with that knowledge base
RPC	CXMLRPC *	Remote procedure call pointer
DBName	CString	String represent the name of the database associated with the knowledge base
DBPath	CString	String represent the full path of the database associated with the knowledge base
m_CKBDBList	CKBDBList	List of database association
m_CTblKeysList	CTblKeysList	List of database tables keys
m_CBDKeysList	CStringArray	List of flat keys
CurrentCpt	CString	String represent the name of the current concept
CurrentProp	CString	String represent the current property
Version	int	The version number of the knowledge base

21-3 Methods

This class has 31 methods

21-3-1Serialize

This method is used to save and load the rule class on hard disk.

Input	Type	Usage
ar	CArchive	Archive class (high level file object)

21-3-2CKBEditorsDoc

Constructor of the CKBEditorsDoc class.

21-3-3~CKBEditorsDoc

Disstructor of the CKBEditorsDoc class.

21-3-4AbductAll

This method fills the temporary concept list with the specific concepts, properties, which are existed in the action of the specific cluster-rules. This action should contains specific value in the input *strVal*

Name	Type	Meaning
strCluster	CString	String represent the cluster name
strCpt	CString	The name of the concept
strProp	CString	The name of the property
strVal	CString	The name of the value

21-3-5AbductAll

This method fills the temporary concept list with the specific concepts, properties, which are existed in the action of the specific cluster-rules. This action should contains specific concept-property in the input strCpt, strProp

Name	Type	Meaning
strCluster	CString	String represent the cluster name
strCpt	CString	The name of the concept
strProp	CString	The name of the property

21-3-6AddinTempList

This Method takes an input the condition of the rule and decomposes that condition into set of concepts, properties, and values, and adds those concepts and properties and values into temporary concept list.

Name	Type	Meaning
str	CString	String represent the condition of the rule

21-3-7AssertToWM

This method add concept property value to working memory, this concept does not exist in the concept list

Name	Type	Meaning
strCpt	CString	The name of the concept
strProp	CString	The name of the property
strVal	CString	The name of the value

21-3-8ClearDoc

Initialize all document lists and working memory

Name	Type	Meaning
strCpt	CString	The name of the concept
strProp	CString	The name of the property
StrVal	CString	The name of the value

21-3-9DecomposeString

This method decompose the input string into the concept and property

Name	Type	Meaning
str	CString	The name of the concept and property with the delimiter
Str1	CString&	The name of the concept
Str2	CString&	The name of the property
delemeter	Const char *	The character by which the string is decomposed into concept and property

21-3-10 DeleteContents

Initialize all document lists and working memory, In addition it initialize the document.

21-3-11 GetVersion

Return with the version of the knowledge base.

21-3-12 SetVersion

Set the version of the knowledge base.

21-3-13 InitWM

Initialize the working memory

21-3-14 IsSourceValue

Test the source of value of a property

Name	Type	Meaning
ssSource	CString	The name of the source of value This may be "Database" or "Relation" etc…
ssCpt	CString	The name of the concept
ssProp	CString	The name of the property

Return true if the source of value of the property match the source in the input otherwise it returns false.

21-3-15 IsSourceValue

An overloaded of the above method, it test the source of value of a property, but the input is different

Name	Type	Meaning
ssSource	CString	The name of the source of value This may be "Database" or "Relation" etc…
Prop	CProperty*	Pointer to the property

21-3-16 OnCloseDocument

The MFC framework calls this event when the document is closed

21-3-17 OnNewDocument

The MFC framework calls this event when new document is opened

21-3-18 OnOpenDocument

The MFC framework calls this event when the document is opened

21-3-19 PlayInference

This method play the inference step in the input parameter

Name	Type	Meaning
strInf	CString	The name of the inference step

21-3-20 RemoveCptPropValFromWM

This method Remove the concept-property-value from the working memory, this method not implemented yet.

Name	Type	Meaning
Cpt	CConcept *	Pointer to the concept
strVal	CString	The value name

21-3-21 RemoveItemFromWM

This method Remove the concept-property from the working memory.

Name	Type	Meaning
Cpt	CString	The concept name
Prop	CString	The property name

21-3-22 PrintWM

This method is used for debugging, it print the working memory list to an HTML file on the current directory of the knowledgebase.

21-3-23 SetDBAssociationToWM

This method gets all association with the database and retrieves its value and set it in the working memory.

21-3-24 SetDBValueToWM

This method set concept property value in the working memory.

Name	Type	Meaning
strCpt	CString	The concept name
strProp	CString	The property name
strVal	CString	The value name

21-3-25 SetDefaultInWM

This method set the default value of all properties in the knowledge base in working memory.

22- Class Name CFunction

22-1 Description:

> *This class is used to store the function structure as a part of the knowledge base*

22-2 Properties:

Name	Type	Meaning
FunID	CString	Function Identification
Output	CString	The output of the function in format concept property
Body	CString	The body of the function
OutputType	CString	The type of the output
pView	CKBEditorsView*	Pointer to the view class
m_Doc	CKBEditorsDoc*	Pointer to the document class

22-3 Methods

This class has 11 methods

22-3-1 Serialize

This method is used to save and load the rule class on hard disk.

Input	Type	Usage
ar	CArchive	Archive class (high level file object)

22-3-2CFunction

Constructor of the CFunctuon class.

Input	Type	Usage
pDoc	CKBEditorsDoc*	Used to init. The pointer to the document when create an instance of the function to make it possible to access on the document class

22-3-3GetBody

Return the body of the function in a string format.

22-3-4GetOutput

Return the output of the function in a string format.

22-3-5GetOutputType

Return the output type of the function in a string format.

22-3-6SetBody

Set the body of the function in the instance.

Input	Type	Usage
Bodystr	CString	The string represent the body of the function

22-3-7SetOutput

Set the output of the function in the instance.

Input	Type	Usage
Outputstr	CString	The string represent the output of the function

22-3-8SetOutputType

Set the output type of the function in the instance.

Input	Type	Usage
OutputstrType	CString	The string represent the type of the output of the function

22-3-9EvalFunction

Evaluate the function and return the result in a string format.

22-3-10 PlayFun

Evaluate the function and set the result in the working memory.

22-3-11 PlayFunList

Evaluate group of functions.

Input	Type	Usage
FunGroup	CObArray *	Array contains the list of functions

23- Class Name CCptPage

23-1 Description:

> *This class provide the interface functionality for editing the concepts*

23-2 Properties:

Name	Type	Meaning
m_CptTree	CTreeCtrl	The variable of a tree control that display a concept list in a hierarchical tree
m_Desc	CString	String variable that hold the description of the concept
m_Doc	CKBEditorsDoc*	Pointer to the document class
m_NameA	CString	String variable that hold the Arabic name of the concept
OldStr	CString	Temporary string to refer to the last name of the concept before editing its name
str1	CString	Temporary string to refer to the string of the tree at edit control when get focus
SetEditFocus	BOOL	Flag to indicate it is in edit mode
SetDescFocus	BOOL	Flag to indicate the description control of the concept when it is receive focus
StatusAdd	BOOL	If true means it came from add event

23-3 Methods

This class has 15 methods

23-3-1 AddChildInTree

This method recursively add the child to the current concept ID in the input. Note: the concept is uniquely identified by its ID

Input	Type	Usage
TreeItem	HTREEITEM	Pointer to the current tree item
ID	Long	The id of the concept

23-3-2AddLegal

This method adds legal values to the specific property.

Input	Type	Usage
ssLegal	CString	String represent legal values of the property Prop delimited by comma
Prop	CProperty*	Pointer to the property to which the legal is added

23-3-3AdjustLegal

This method is used when importing from KROL file into KSR knowledge base, take in input string in a KROL format and decompose it into property name and set of legal values separated by comma.

Input	Type	Usage
str	CString	String represent legal values of the property in a KROL format
ssProp	CString&	The name of the returned property
ssLegal	CString&	The returned legal values separated by comma

23-3-4AdjustPrompt

This method is used when importing from KROL file into KSR knowledge base, take in input string in a KROL format and decompose it into property name and prompt of that property.

Input	Type	Usage
str	CString	String represent the prompt of the property in a KROL format
ssProp	CString&	The name of the returned property
ssPrompt	CString&	The returned prompt

23-3-5AdjustProperty

This method is used when importing from KROL file into KSR knowledge base, take in input string in a KROL format and return the property name from it.

Input	Type	Usage
str	CString	String represent the prompt of the property in a KROL format
Return Type	**Description**	
CString	The name of the property	

23-3-6AdjustType

This method is used when importing from KROL file into KSR knowledge base, take in input string in a KROL format and return the type of the property.

Input	Type	Usage
str	CString	String represent the type of the property in a KROL format
ssProp	CString&	The name of the returned property
ssType	CString&	The returned type

23-3-7GetConcept

This method is used when importing from KROL file into KSR knowledge base, take in input string in a KROL format and return the concept in KSR representation.

Input	Type	Usage
str	CString	String represent the type of the property in a KROL format
Return Type	**Description**	
CConcept *	Pointer to KSR concept class	

23-3-8GetProp

This method is used when importing from KROL file into KSR knowledge base, take in input string in a KROL format and return the name of the property.

Input	Type	Usage
str	CString	String represent the property in the KROL format
Return Type	**Description**	
long	The id of the concept	

23-3-9GetSuperID

This method is used when importing from KROL file into KSR knowledge base, take in input string in a KROL format and return the ID of the concept.

Input	Type	Usage
str	CString	String represent the super statement in the KROL format
ssProp	long	The returned property name

23-3-10 DeleteTreeItems

This method is used to remove the concept from the tree control

Input	Type	Usage
TreeItem	HTREEITEM	Pointer to the current tree item selection

23-3-11 DrawCptTree

This method is used to redraw the entire concept list in a tree control

23-3-12 GetInputRole

This method is used get the input role of specific rule condition.

Input	Type	Usage
str	CString	The condition of the rule
InRole	CString	Constructed input role
Return Type	**Description**	
CString	String represent the input role of that condition in an HTML format	

23-3-13 GetOutputRole

This method is used get the output role of specific rule action.

Input	Type	Usage
str	CString	The action of the rule
outRole	CString	Constructed output role
Return Type	**Description**	
CString	String represent the output role of that action in an HTML format	

23-3-14 SeeDoc

This method used to initiate the pointer to the document class when creating concept page

Input	Type	Usage
Doc	CKBEditorsDoc*	Pointer to the document class

23-3-15 Search

This method is used get the search in the tree for specific concept and set the focus to that concept if it is exist in the tree

Input	Type	Usage
str	CString	The name of the concept
TreeItem	HTREEITEM	Pointer to the node by which the search is start
Return Type	**Description**	
HTREEITEM	Pointer to the founded concept node	

24- Class Name CTable

24-1 Description

> *This class is used to store the table structure as a part of the knowledge base*

24-2 Properties:

Name	Type	Meaning
Body	CBody	This variable holds the body of the table as two dimensional array
MaxRows	int	This variable holds the maximum row that table currently hold.
ResultList	CResult	Vector that holds result associated with each row
Schema	CStringArray	Array of input cpt-prop of table

24-3 Methods

24-3-1 CTable

This method is a constructor of CTable just to give MaxRow initial value equal to one.

24-3-2 GetCell

This method returns Cell of table in specific row and col.

Input	Type	Usage
Row	int	Integer value represent row
Col	int	Integer value represent column

24-3-3 SetCell

This method is used when importing from KROL file into KSR knowledge base, take in input string in a KROL format and decompose it into property name and set of legal values separated by comma.

Input	Type	Usage
Row	int	Integer value represent row
Col	int	Integer value represent column
OPVal	CString	String value will be set in above row and column

24-3-4 Play

This method is used in reasoning to parse the table and then the result of successful row in working memory.

Input	Type	Usage
M_WM	CWMStructure*	Pointer to working memory to be searched
pCptList	CCptList *	Pointer to the concept list to be used for knowing property type

24-3-5 Serialize

This method is used to save and load the CTable class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

25- Class Name Ccell

25-1 Description

> *This class represents a basic element of CTable class. It is like a premises in rule eg a > 3.*

25-2 Properties:

Name	Type	Meaning
Attribute	CString	Concatenations of Cpt-Prop
Operator	CString	Logical operator eg <,<=,=, etc..
Value	CString	Right hand side of the premise represented as literal value.

25-3 Methods

25-3-1 Serialize

This method is used to save and load the CCell class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

26- Class Name CimageOb

26-1 Description

> *This class all multimedia file name*

26-2 Properties:

Name	Type	Meaning
m_ImageList	CStringList	List of strings for all multimedia file name

26-3 Methods

26-3-1 Serialize

This method is used to save and load the ImageList class on hard disk.

Input	Type	Usage
Ar	CArchive	Archive class (high level file object)

27- Class Name CtreeItem

27-1 Description

> *This class to facilitate search on tree*

27-2 Properties:

Name	Type	Meaning
TreeItem	HTREEITEM	Represent a tree item

28- Class Name CTableDlg

28-1 Description:

> *This class is used to display the Dialogue to CTable class. It is considered as table editor.*

28-2 Properties:

Name	Type	Meaning
M_Date	CTime	Member variable associated with date control used to hold the value the property date
m_Prompt	CString	Member variable associated with textbox control used to hold the prompt of the property
Prop	CProperty *	Pointer to the CProperty object
Val	CString	Store the acquired date into the string format

28-3 Methods

This class has five methods

28.3.1 CDateDlg

Constructor of the dialogue

Input	Type	Usage
pParent	CWnd*	Pointer to the parent window (default NULL)

28.3.2 DoDataExchange

This method created by the framework of MFC weathered, is used to make an association between the visual interface appear to the user and member variables

Input	Type	Usage
pDX	CDataExchange*	Pointer to the data exchange class

28.3.3 OnInitDialog

This event handler is called when the dialogue appear on the screen, used to init the member variables of the dialogue class

Return Type	Description
BOOL	True if the dialogue appear successfully, otherwise false.

28.3.4 SeeData

This method is used to init the CProperty pointer that is used by dialogue to retrieve the prompt of that property.

Input	Type	Usage
inProp	CProperty *	Pointer to the CProperty class

28.3.5 OnOK

This event handler is used to close the dialogue and store the value gets from the user in the table structure.

29- Class Name AddDeleteDlg

29-1 Description:

> *This class is used to display the Dialogue to build table schema.*

29-2 Properties:

Name	Type	Meaning
m_ConceptLstCtrl	CListBox	Member variable used for holding the concepts name.
FlxResultChanged	BOOL	Not used
m_Doc	CKBEditorsDoc *	Pointer to the document class
m_PropertyLstCtrl	ClistBox	Member variable used for holding the properties name.
m_Result	Cstring	Member variable used for holding the results of table or the output its constructed as concept-property string.
m_ValueLstCtrl	ClistBox	Member variable associated with List box control used to hold the list of input of table
TableObj	CTable *	Poiter to table object.

29-3 Methods

This class has four methods

29.3.1 AddDeleteDlg

Constructor of the dialogue

Input	Type	Usage
pParent	CWnd*	Pointer to the parent window (default NULL)

29.3.2 DoDataExchange

This method created by the framework of MFC weathered, is used to make an association between the visual interface appear to the user and member variables

Input	Type	Usage
pDX	CDataExchange*	Pointer to the data exchange class

29.3.3 OnInitDialog

This event handler is called when the dialogue appear on the screen, used to init the member variables of the dialogue class

Return Type	Description
BOOL	True if the dialogue appear successfully, otherwise false.

29.3.4 OnOK

This event handler is used to close the dialogue and store the value gets from the user in the table structure.

29.3.5 SeeDoc

This method used to initiate the pointer to the document class when creating concept sheet

Input	Type	Usage
Doc	CKBEditorsDoc*	Pointer to the document class

29.3.6 OnAddResult

This method used for adding results to Table object constructed from m_ConceptLstCtrl control and m_PropertyLstCtrl control.

29.3.6 OnAddSchema

This method used for adding input to Table object constructed from m_ConceptLstCtrl control and m_PropertyLstCtrl control.

30- Class Name CTableNameDlg

30-1 Description:

> *This class is used to display the Dialogue to accept table name for both new table or rename old table.*

30-2 Properties:

Name	Type	Meaning
strTableName	CString	Member variable used for holding the value of table name.
OpStr	CString	Member variable used for holding the value of operation type if its new table name or renaming old table.

30-3 Methods

This class has four methods

30.3.1 CTableNameDlg

Constructor of the dialogue

Input	Type	Usage
pParent	CWnd*	Pointer to the parent window (default NULL)

30.3.2 DoDataExchange

This method created by the framework of MFC weathered, is used to make an association between the visual interface appear to the user and member variables

Input	Type	Usage
pDX	CDataExchange*	Pointer to the data exchange class

30.3.3 OnInitDialog

This event handler is called when the dialogue appear on the screen, used to init the member variables of the dialogue class

Return Type	Description
BOOL	True if the dialogue appear successfully, otherwise false.

30.3.4 OnOK

This event handler is used to close the dialogue and store the value gets from the user in the table structure.

31- Class Name OperationVlaueDlg

31-1 Description:

> *This class is used for display the Dialogue to accept table cell value from user.*

31-2 Properties:

Name	Type	Meaning
Attribute	CString	Member variable used for holding the variable of associated with column of table it's constructed as concept-property string.
FromResult	CString	Member variable used for holding the value of the Attribute Object of a specific cell in table

		object.
strIn	CString	Member variable used for holding the stored value of Attribute object
strOut	CString	Member variable used for holding the changed value of Attribute object to be assigned in its table cell
m_Doc	CKBEditorsDoc *	Pointer to the document class
m_OpListCtrl	CComboBox	Member variable used for holding all possible operators that Attribute object can use.
M_ValueListCtrl	CComboBox	Member variable used for holding all possible values of that Attribute object can hold.

31-3 Methods

This class has four methods

31.3.1 OperationVlaueDlg

Constructor of the dialogue

Input	Type	Usage
pParent	CWnd*	Pointer to the parent window (default NULL)

31.3.2 DoDataExchange

This method created by the framework of MFC weathered, is used to make an association between the visual interface appear to the user and member variables

Input	Type	Usage
pDX	CDataExchange*	Pointer to the data exchange class

31.3.3 OnInitDialog

This event handler is called when the dialogue appear on the screen, used to init the member variables of the dialogue class

Return Type	Description
BOOL	True if the dialogue appear successfully, otherwise false.

31.3.4 OnOK

This event handler is used to close the dialogue and store the value gets from the user in the table structure.

31.3.5 GetProp

This method get the part of property from constructed string in the format `Concept-Property` i.e. this function split the string into two parts delimited by a specific char and return the second part.

Input	Type	Usage
Attr	CString	string in the format `Concept-Property`
A	Char	Delimiter character.

31.3.6 GetCpt

This method get the part of concept from constructed string in the format `Concept-Property` i.e. this function split the string into two parts delimited by a specific char and return the first part.

Input	Type	Usage
Attr	CString	string in the format `Concept-Property`
A	Char	Delimiter character.

31.3.6 SeeDoc

This method used to initiate the pointer to the document class when creating concept sheet

Input	Type	Usage
Doc	CKBEditorsDoc*	Pointer to the document class

32- Class Name FunToken

32-1 Description:

> *This class used to divide the function stream to token.*

32-2 Properties:

Name	Type	Meaning
StartPos	Static int	The position in the function stream
LineNumber	Static int	The line number of the function stream

32-3 Methods

This class has 9 methods

32-3-1 GetStartPos

Get the start position in the function stream (position of the pointer)

Return Type	Description
int	The start position in the stream of the function

32-3-2 GetLineNumber

Get line number in the function text

Return Type	Description
int	Line number in the function

32-3-3 ReturnBack

Return the pointer one step to back and the length of this step equal the length of last token.

Input	Type	Usage
Token	CString	The last token to compute its length and return the pointer in function stream

32-3-4 Reset

This function reset all Properties in this class

32-3-5 GetToken

This function is the most important function in this class, this function Return the token string from the function stream

Input	Type	Usage
FunStream	CString	The stream of the function
Token	CString	The token that return from the function

Return Type	Description
BOOL	If the function successes to get token return true else return false

32-3-5 GetConcept

Get token from function stream and check if it Concept or not

Input	Type	Usage
FunStream	CString	The stream of the function
Token	CString	The token that return from the function

Return Type	Description
BOOL	Return true if the token is Concept else return false

32-3-7 GetProp

Get token from function stream and check if it Property or not

Input	Type	Usage
FunStream	CString	The stream of the function
Token	CString	The token that return from the function

Return Type	Description
BOOL	Return true if the token is Property else return false

32-3-8 GetDate

Get token from function stream and check if it Date or not

Input	Type	Usage
FunStream	CString	The stream of the function
Token	CString	The token that return from the function

Return Type	Description
BOOL	Return true if the token is Date else return false

32-3-9 WriteToken

Get all tokens from function and write it to file

Input	Type	Usage
FunStream	CString	The stream of the function

33- Class Name FunParser

33-1 Description:

> *This class used in parse the function stream then show if the statement correct or not*

33-2 Properties:

Name	Type	Meaning
TokenObj	FunToken	The position in the function stream
OutputType	Int	The data type of the output of function
BodyType	Int	The data type of the body of the function
MathBuiltFun	Int	The data type of Mathematical function
F	static int	The last state in the parsing (state of Finite State Machine FSM)

33-3 Methods

This class has 7 methods

33-3-1 ParseHeader

Parse the head of the function

Input	Type	Usage
FunStream	CString	The stream of the function
Error	CString	If function fail in parsing the error return on it
pCptList	CCptList	To sure the concept exist in the concept list

Return Type	Description
BOOL	If the function successes to parse the header return true else return false

33-3-2 ParseBody

Parse the body of the function

Input	Type	Usage
FunStream	CString	The stream of the function
Error	CString	If function fail in parsing the error return on it
pCptList	CCptList	To sure the concept exist in the concept list

Return Type	Description
BOOL	If the function successes to parse the body return true else return false

33-3-3 Reset

This function reset all Properties in this class

33-3-4 Parse_CPT_PROP

This function parse (Copcept..Property) as one token

Input	Type	Usage
FunStream	CString	The stream of the function
Cpt	CString	The name of concept
Prop	CString	The name of property
Error	CString	If function fail in parsing the error return on it

Return Type	Description
BOOL	If the function successes to parse Concept Property then return true else return false

33-3-5 ParseBuilt

This function parses the built in function such as abs, sqrt, day, now, …

Input	Type	Usage
FunStream	CString	The stream of the function
Fun	CString	The name of built in function
Error	CString	If function fail in parsing the error return on it
pCptList	CCptList	To sure the concept exist in the Concept list
Return Type	**Description**	
BOOL	If the function successes to parse built in function then return true else return false	

33-3-6 ParseMath

The function complete the parsing of the built in function

Input	Type	Usage
FunStream	CString	The stream of the function
Error	CString	If function fail in parsing the error return on it
pCptList	CCptList	To sure the concept exist in the Concept list

Return Type	Description
BOOL	If the function successes to parse built in function then return true else return false

33-3-7 ParseDate

This function parses the format of date

Input	Type	Usage
FunStream	CString	The stream of the function
Error	CString	If function fail in parsing the error return on it
pCptList	CCptList	To sure the concept exist in the Concept list
Return Type	**Description**	
BOOL	If the function successes to parse date then return true else return false	