

DOCUMENT STORAGE SYSTEMS, INC.

DSIO – oCNTs

(Custom Note Templates)

Development/Technical Guide

Theodore Fontana

4/22/2014

Date	Revision	Description		Author
3/27/14	S7 B6	oCNTFramework.bpl (Package)	V0.0.2.2	Theodore Fontana
		ToForm (Component)	V0.0.2.2	
		uExtndComBroker (unit)	V0.0.2.2	
		CPRSNoteCOM.dll (COMObject)	V0.0.6.0	
		oCNTDialogs.dll (OCNT Extension)	V0.0.1.2	
		oCNTFrame.dll (OCNT Test Rig)	V0.0.1.2	
		OB H&P Initial OCNT (OCNT_OBHP.dll)	V1.0.0.0	
		Updates to the DSIO OCNT DIALOGS and DSIO OCNT CONFIGURATION files – data has been added for OB H&P Initial.		
3/21/14	S7 B5	oCNTFramework.bpl (Package)	V0.0.2.1	Theodore Fontana
		ToForm (Component)	V0.0.2.1	
		uExtndComBroker (unit)	V0.0.2.1	
		CPRSNoteCOM.dll (COMObject)	V0.0.5.3	
		oCNTDialogs.dll (OCNT Extension)	V0.0.1.1	
		oCNTFrame.dll (OCNT Test Rig)	V0.0.1.1	
		- ReportOrder now only reorders items with the ok button - License information added to source - udlgOBSpread converted - Removed the DSIO CREATE DIALOG RPC and added the functionality to DSIO OCNT DIALOGS - Corrected the entries in DSIO OCNT DIALOGS - Added support to Comboboxes in DSIO OCNT CONFIGURATION - Removed the DSIO OCNT DIALOGS RPC and created DSIO OCNT POPULATE to handle any control needing to be populated from VistA - Currently supports TListBox and TComboBox - Controls populated from VistA with startup values - Supported: TListBox (Dialogs), TComboboxes - Added support for save, restore, and note generation to TComboBox,TListBox - ReportCollection Items can no longer have their OwningObject changed - Cleanedup extra spaces in note - Initial Configuration Data added to note so every extended item can hold data brought in from VistA for initial display (this is not restore or user entered data - this is more on the lines of options in lists that the user would select) - NoteItem Dialog Return can now only be set as a TMemo or TRichEdit - A message now displays if there is any values from VistA used to restore the OCNT. The user must click "Yes" to restore this values. - Dialogs can now save discreet data		

		<ul style="list-style-type: none">- The user will get a message if values of the dialog exists and to click "Yes" in order to restore them.- Added TjvStringGrid to supported Save and Restore Components- Updated the DSIO OCNT CONFIGURATION file to store multi and single valued controls<ul style="list-style-type: none">- The creation of the OCNT populates these controls with the values<ul style="list-style-type: none">- VistA should always have the control name as uppercase - for lookup puposes- Activated the CPRS TimeOut Timer- Added an AutoSave Timer- Updated the DSIO OCNT DIALOGS file to act as a configuration file- Created the RPC: DSIO OCNT DIALOG POPULATE that mimics the DSIO OCNT POPULATE- Removed the Populate from VistA property from ReportCollectionItems and extended the ability to all components on the OCNT form- Some dialogs use other dialogs and in this case they won't have their IEN set<ul style="list-style-type: none">so a RPC (DSIO OCNT DIALOG LOOKUP) was created and added to the DSIO OCNT CONTEXT- Dialogs (Show and ShowModal) both will populate a Dialog from VistA- Corrected Discreet Data TCheckBox and TRadioButton - unchecking is now treated as a value so a change from Checked to unChecked will now restore correctly- VistA configurable control fields have been transformed to uppercase to ensure proper lookup- Only control with data not matching VistA configuration data will be collected to save discreetly- The Discreet Data file system will not recognize Job IDs and will overwrite existing entries of the same job rather than creating new ones- Auto save has been disabled - it was causing some problems - a solution possible for future use		
3/12/14	S7 B2	oCNTFramework.bpl (Package)	V0.0.2.0	Theodore Fontana
		ToForm (Component)	V0.0.2.0	
		uExtndComBroker (unit)	V0.0.2.0	
		CPRSNoteCOM.dll (COMObject)	V0.0.5.2	
		oCNTDialogs.dll (OCNT Extension)	V0.0.1.0	
		oCNTFrame.dll (OCNT Test Rig)	V0.0.1.0	
		Initial Release of OCNT.		

Contents

Purpose	4
Components - VistA	5
Components - GUI.....	7
OCNT Architecture	8
Features and Functionality	10
Client Side	10
Development Side.....	10
OCNT Installation	12
Registering the COMObject	13
For Local Installations:	13
For Network Installation:	13
VistA Configuration	15
Create an entry in the OE/RR COM OBJECTS file... ..	15
Setting Required Parameters... ..	15
Setting the Menu Context... ..	16
CPRS Configuration	17
Remote Procedure Calls.....	19
DSIO OCNT GET LOCATION	19
DSIO OCNT DIALOG LOOKUP	20
DSIO OCNT STORE	21
DSIO OCNT POPULATE	24
DSIO OCNT DIALOG POPULATE.....	26
DSIO OCNT RESTORE.....	28
Development.....	29
ReportCollection	33
ReOrder ReportCollection Items:.....	34
Title, BeforeValue, and AfterValue	36
PopulatefromVistA.....	36
Saving Discreet Data	39
Additional Reading.....	42

Purpose

Maturity Tracker oCNTs or Custom Note Templates are designed to aid a user in providing an abundance of relevant information to generate a note. Essentially, an OCNT is an advanced Reminder Dialog.

The components and completed oCNTs for DSIO are not to be confused with their proprietary counterparts. Much of the code has been rebuilt and redesigned. However the biggest differences a user will see is that these oCNTs utilize parameters and VistA files instead of the host's file system; additionally, each element of the OCNT has the potential to be saved discreetly. From a developer's standpoint the foundation of these CNTs is built from the ToForm component which allows the developer to extend any Window's based control dropped onto the form to interact with VistA and be part of the generated note.

This document will cover everything from configuration to development.

Components - VistA

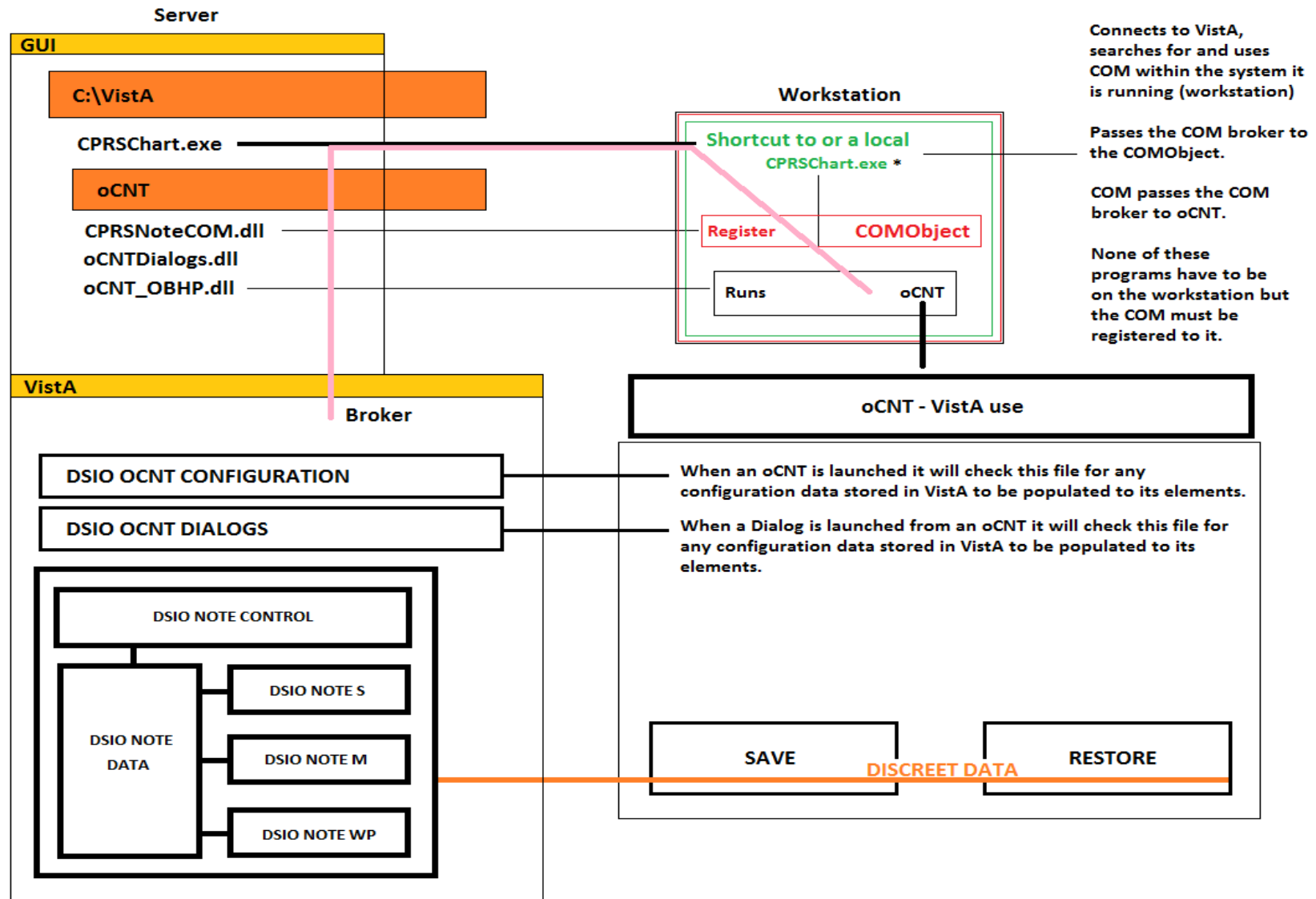
Type	Name	Purpose
Menu Context		
	DSIO OCNT CONTEXT	This is the menu context that must be assigned to a user in order to use the RPCs needed to run an OCNT.
Parameter		
	DSIO OCNT LOCATION	This parameter allows a site to control the root location of an oCNTs. For example, if the oCNTs are stored on C:\oCNTs\ then this will be the value of this parameter. The oCNTs can then either be stored directly in this folder or within a subfolder(s) but this additional distinction must be the CPRS template's passed value – either the name of the OCNT if not subfolder or subfolder\name of OCNT.dll. It is important to note that the parameter value and the template's passed value must create a valid location (including file) when combined. This parameter also allows for easy migration (as long as the subfolder structure remains the same) changing from C:\oCNTs\ to another root means resetting this parameter's value.
Routine		
	DSIO0	This routine hosts the initial M code for these oCNTs.
RPC		
	DSIO OCNT GET LOCATION	This RPC gets the value of the DSIO OCNT LOCATION parameter.
	DSIO OCNT DIALOG LOOKUP	This RPC returns an OCNT Dialog's IEN by classname lookup.
	DSIO OCNT STORE	This RPC will store discreet data for an OCNT or an OCNT dialog.
	DSIO OCNT POPULATE	This RPC pulls configuration information from VistA by TIU Document Title and OCNT component and returns the values to be initially populated to it.
	DSIO OCNT DIALOG POPULATE	This RPC preforms the same purpose as DSIO OCNT POPULATE but for OCNT dialogs.
	DSIO OCNT RESTORE	This RPC behaves like the population RPCs but instead it populates OCNT components with discreet data saved by TIU Document Title.

Files		
	DSIO NOTE CONTROL	Please see DDCS.
	DSIO NOTE DATA	Please see DDCS.
	DSIO NOTE S	Please see DDCS.
	DSIO NOTE M	Please see DDCS.
	DSIO NOTE WP	Please see DDCS.
	DSIO OCNT CONFIGURATION	This file stores some configuration details based on a TIU Document Title such as populating ListBoxes with dialog titles.
	DSIO OCNT DIALOGS	This file stores the display, form, and class name of a dialog as well as configuration information used by the GUI to create and display a dialog.

Components - GUI

Type	Name	Purpose
Delphi Packages		
	oCNTFramework.bpl	This package contains the OCNT component and extended broker unit.
Delphi Units		
	oCNTBase	This unit is the main OCNT package unit and must be listed within the interface uses clause of the form containing the ToForm.
	uExtndComBroker	This unit transforms the CPRS ICPRSBroker to behave like ORNet. This unit has to be added to a project in the interface clause in order to access the broker.
Delphi Components		
	ToForm	This is the only OCNT component. It controls all controls dropped onto it to generate a note, save values discreetly, and provides the look, feel, and functionality of an OCNT.
COMObject		
	CPRSNoteCOM.dll	This COMObject is required to launch an OCNT. It passes TIU information to the OCNT as well as referencing its broker.
OCNT Extensions		
	oCNTDialogs.dll	This DLL contains all common dialogs used between oCNTs. An OCNT must load this DLL in order to use these forms.
CNTs		
	Initial (OB H&P Initial)	
	Follow Up (OB H&P FU)	
	Nurse Postpartum	

oCNT WITH DDCS



The DSIO oCNTs can only be run through the CPRS TIU Note ComObject functionality and so cannot run standalone nor can they access Vista without it being through CPRS. The OCNT uses ICPRSBroker which is set up in CPRSNoteCOM.dll using uExtndComBroker.pass to behave like ORNet before it is passed to the called OCNT. Before it is passed to the called OCNT the ComObject will use it to retrieve the OCNT root path using the RPC DSIO OCNT GET LOCATION which uses the parameter DSIO OCNT LOCATION. Once it has this value it will concatenate it with the passed value to find and launch the called OCNT.

Once an OCNT is launched the OCNT will loop through all of its owned components and use the RPC - DSIO OCNT POPULATE to lookup any of these components by name and TIU Document IEN which in turn is used to check the DSIO OCNT CONFIGURATION file to look under for the TIU Document Title and if in the file will check if the component name also exists and if so, will retrieve any associated value(s). If the RPC returns a value it will be used to set the component. The idea here is to allow OCNT selection type components to be changed by a site by site basis without the need for redevelopment. This same functionality is performed by DSIO OCNT DIALOG POPULATE when a dialog is opened but uses the DSIO OCNT DIALOGS file.

When an OCNT is closed, finished, or saved it will loop through all of its ReportCollection Items and build an array and once finished it will pass this array to Vista using the RPC - DSIO OCNT STORE which will then save the data associated by component and TIU Document discreetly. The APIs and files used for this process is part of the DSIO - DDCS (Discreet Data Control and Storage) package; for more information please refer to the DSIO_DDCS.pdf document. Likewise, when an OCNT dialog is OK'ed it will loop through all of its owned components and will use the same RPC to store all of its components and their values discreetly.

Features and Functionality

Client Side

CPRS Friendly

- 1 CPRS is still accessible when an OCNT is active.
- 2 OCNT Broker usage recorded in CPRS Broker History.
- 3 Only one OCNT can be active from any one CPRS session at any given time.
- 4 The OCNT will close itself upon CPRS timeout. (Incomplete)

Configuration

- 5 Each OCNT (by TIU Document Title) and OCNT Dialog can be configured in VistA to populate its components when first being activated.

Save and Restore

- 6 Clicking Save, Cancel, or Finish will save each data element discreetly within the VistA DDCS control files. An OCNT dialog must be closed using the OK button in order to save.
 - I. The TIU Document Title must be in DSIO NOTE CONTROL (19641.4) file - this should be true for all oCNTs (an OCNT dialog does not have to be saved to this file).
 - II. The discreet data is saved to DSIO NOTE DATA (19641.41) under the TIU Document IEN.
 - III. The actual data is saved in three files depending on their type.
 - a. DSIO NOTE S (19641.43) Single Value
 - b. DSIO NOTE M (19641.44) Multiple Values
 - c. DSIO NOTE WP (19641.45) Word Processing
 - IV. The VistA option – DSIO SHOW DISCREET DATA – will print to the terminal all saved entries associated to a TIU Document (not for use with OCNT dialogs).
- 7 Inform the user that saving was successful or not. (Incomplete)
- 8 When an OCNT or OCNT Dialog is activated it will check to see if that TIU Document has any discreet data saved and if a OCNT Dialog it will check if that TIU Document has that dialog already saved, and if so it will restore the values to the form, but not without first asking the user if they want to restore.

Development Side

- 1 One component (ToForm) to control them all.
 - I. Feel free to use many of Delphi's native components and Jedi's TJsStringGrid.
 - II. Dropping these components on the ToForm will create a ReportCollectionItem (ReportItem) that will allow the developer to easily extend any of these components to be compatible with the oCNTs note.
 - III. This extension cannot be generated when a component is dropped onto a control such as a TPanel or TGroupBox, but a ReportItem can be manually created and associated to the object.
 - IV. ReportItem supports a prefix, suffix, and a title that will be printed to the note it supplied.

- V. A ReportItem can also be made required which will force the user to supply a value to the owning object before generating the note.
 - VI. A ReportItem can be hidden from the note and prevented from being saved discreetly.
 - VII. The ReportItem also has a property called "VistALabel" which will be stored as the field value in the control files – this is to aid in identifying an element if there is need to push the information from the control files elsewhere in VistA.
 - VIII. Lastly, DialogReturn is for TListBoxes only and is used for ListBoxes intending to display dialogs. When a dialog is activated from a TListBox it is expecting to return its note information to a TMemo or TRichEdit – this is what this value is for and can only be populated by either of the two.
- 2 uExtndComBroker allows the developer to use methods like CallV, sCallV, and tCallV as they would in ORNet.

OCNT Installation

For local or network installations preform the following steps.

First, locate the directory where CPRS is installed which should be the top level of your C or other drive on your system under the VistA folder. If there is to top level VistA folder such as C:\VistA\ (D:\VistA) then look in the Program Files (x86) directory (C:\Program Files (x86)\VistA)

Now that you have located the VistA folder, create a directory called “OCNT” (or whatever you prefer but for the purpose of these instructions I will be referencing “OCNT”). In this folder move the following two required DLLs.

- CPRSNoteCOM.dll (this is the COMObject – it handles communication between CPRS and the OCNT)
- oCNTDialogs.dll (this is an extension for the oCNTs – it contains and thus, allows for all oCNTs to use common dialogs) – only one copy of oCNTDialogs.dll can be loaded and it is based upon the below priorities:
 - This extension can be stored in the same folder as your OCNT(s) (Priority 1)
 - In a folder called “Extensions” within the folder as your OCNT(s) (Priority 2)
 - Or in a folder called “Extensions” one level up from your OCNT(s) (Priority 3)

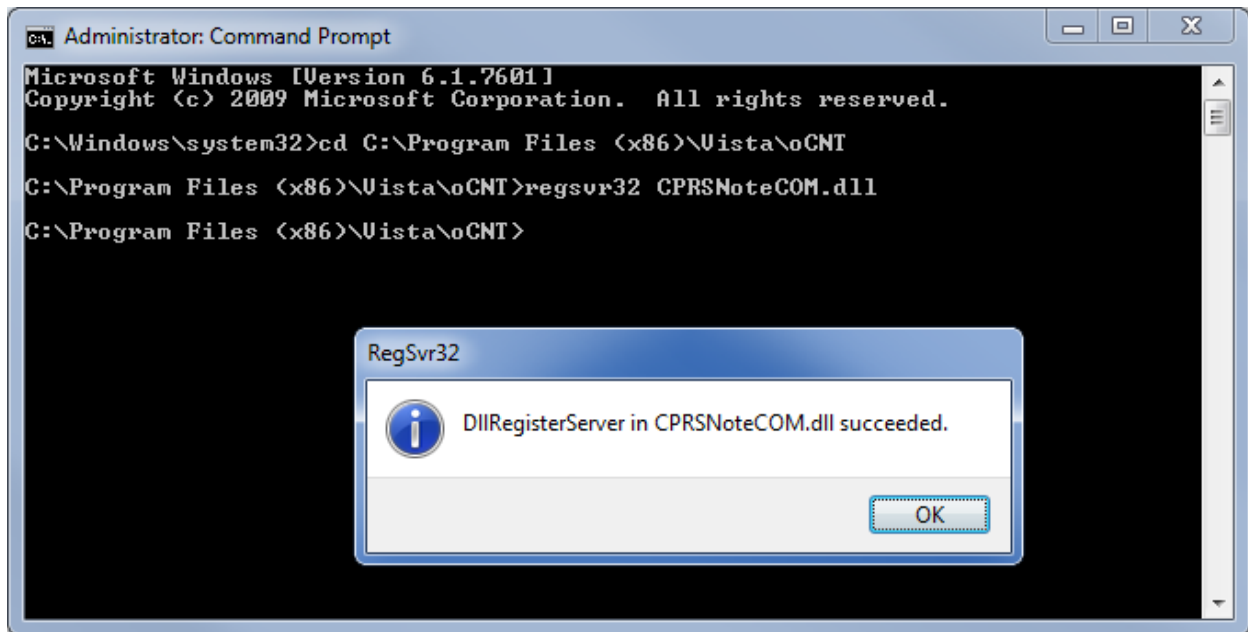
Registering the COMObject

Find your Command Prompt program and right click and “Run as administrator”.

For Local Installations:

Navigate to the directory where you placed the CPRSNoteCOM.dll by using the drive letter followed by colon to change drives and cd to change directories.

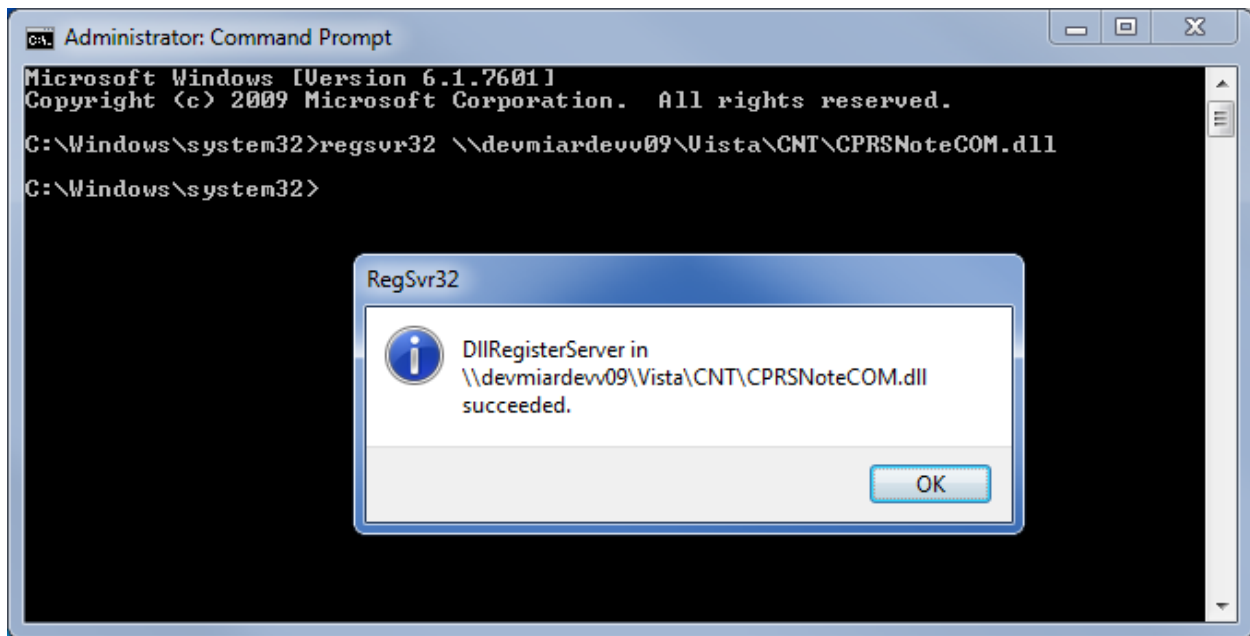
Once the command prompt indicates your directory use the command regsvr32 followed by CPRSNoteCOM.dll to register the COMObject.



For Network Installation:

**Do not attempt to register CPRSNoteCOM.dll both locally and on your network on the same workstation.

Use the command regsvr32 and full network path including CPRSNoteCOM.dll to register the COMObject to your workstation.



*It is important to note that registering a COMObject registers that COMObject to your workstation even if that COMObject is not located on it. Your server, the machine where CPRS, the COMObject, and the oCNTs are located, may already have the COMObject registered but that only applies to that machine; therefore, you must be logged onto that server and running CPRS directly to utilize that registered COMObject.

Vista Configuration

Create an entry in the OE/RR COM OBJECTS file...

If not already supplied as an update, please access FileMan and complete the following minimum required steps to create an entry in the OE/RR COM OBJECTS file:

Example:

VAH>D Q^DI

VA FileMan 22.0

Select OPTION: ENTER OR EDIT FILE ENTRIES

INPUT TO WHAT FILE: OE/RR COM OBJECTS//

EDIT WHICH FIELD: ALL//

Select OE/RR COM OBJECTS NAME: DSIO COMOBJECT

NAME: DSIO COMOBJECT//

OBJECT GUID: {902E9DDC-1479-4D38-8152-481413258BDC}

*The name of this entry just aids the user in identifying the correct COMObject to link to a note template but the OBJECT GUID must be exactly as you see it including the brackets.

Setting Required Parameters...

The purpose of DSIO OCNT LOCATION is to identify the location of the OCNT extension identified as the oCNTDialogs.dll.

Examples:

For Local Installations: C:\Program Files (x86)\Vista\OCNT\

For Network Installations: \\devmiardevv09\Vista\OCNT\

VAH>D ^XPAREDIT

--- Edit Parameter Values ---

Select PARAMETER DEFINITION NAME: DSIO OCNT LOCATION DSIO OCNT Network Address

DSIO OCNT LOCATION may be set for the following:

3 Division DIV [choose from INSTITUTION]

6 System SYS [SMA.FO-ALBANY.MED.VA.GOV]

Enter selection: 6 System SMA.FO-ALBANY.MED.VA.GOV

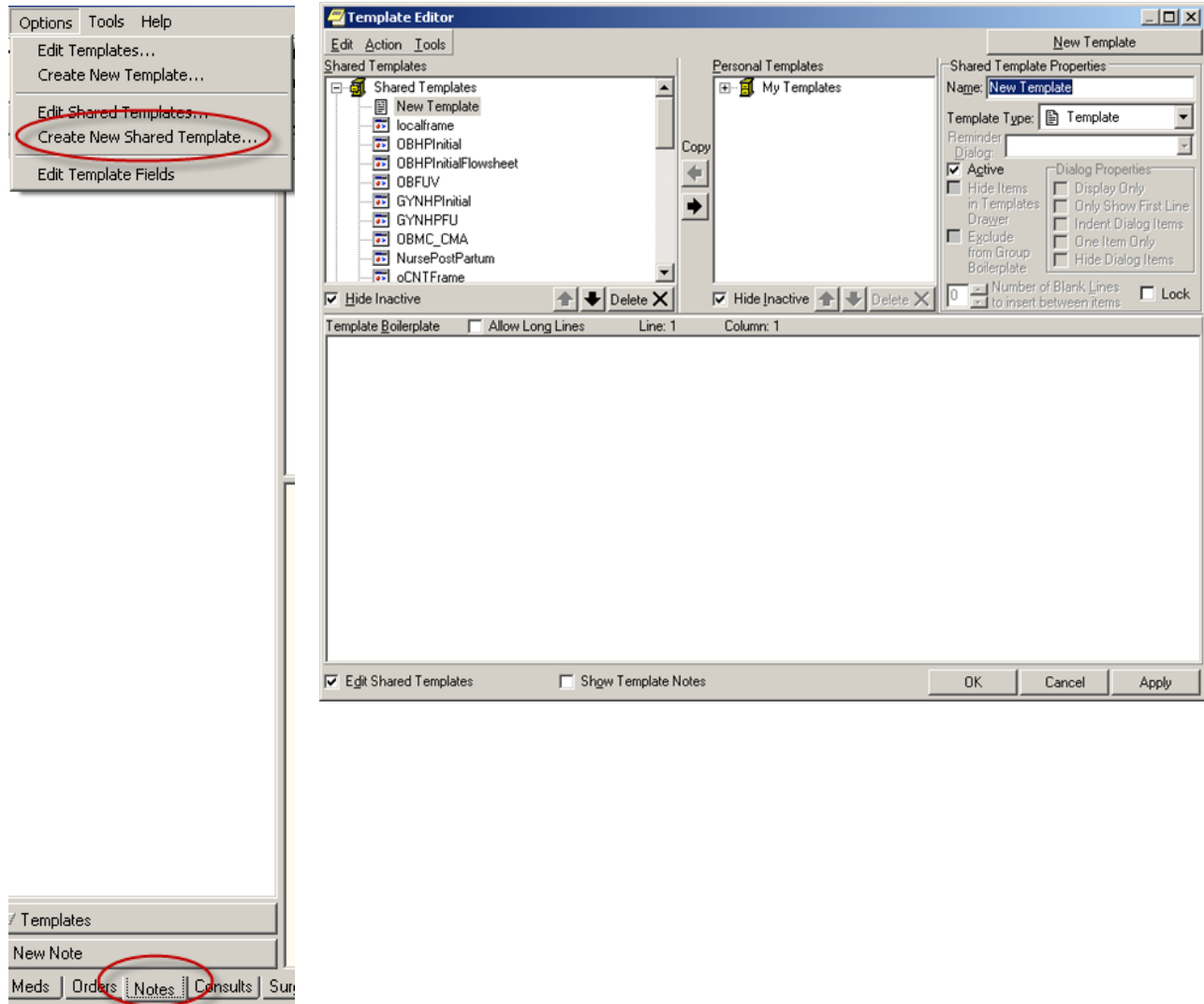
----- Setting DSIO OCNT LOCATION for System: SMA.FO-ALBANY.MED.VA.GOV ----- ADDRESS: \\devmiardevv09\Vista\OCNT\

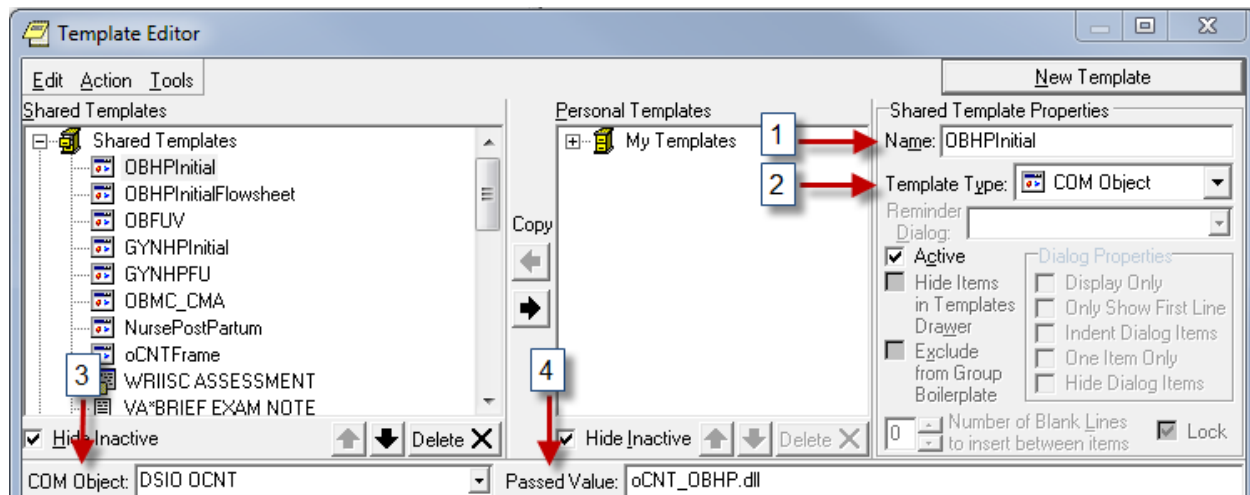
Setting the Menu Context...

The DSIO OCNT CONTEXT option has to be set as one of the user's secondary menu options in order to run an OCNT. You can do this either through FileMan in the NEW PERSON (200) file or through the "User Management" – "Edit an Existing User" menu options.

CPRS Configuration

To create a COMObject based template, access the Template Editor through CPRS via the “Create New Shared Template...” option under “Options” accessible within the “Notes” tab.





1. Set the Name of your Template.
2. Select within the Template Type drop down "COM Object".
3. Select within the COM Object drop down your OE/RR COM OBJECTS file entry you created for the Vista side configuration.
4. Set this field with the complete address including the name of the OCNT DLL of where this particular OCNT is stored.
5. And, of course, hit "OK" or "Apply" and "OK" to close save your changes and close the form when you're done.

*Please Note that each OCNT would have an entry like this using the same COM Object (do not create additional OE/RR COM OBJECTS file entries) with the only difference being the name and passed value.

Remote Procedure Calls

The following RPCs were designed specifically for DSIO oCNTs, for other DSIO RPCs please refer to the documentation DSIO CORE.

DSIO OCNT GET LOCATION

Reference Type

Category OCNT

Description This RPC gets the value of the DSIO OCNT LOCATION parameter.

Format PARAM^DSIO0 (.RET)

Input Parameters

Output	SINGLE VALUE
Success	Value
Failure	null

PARAM^DSIO0(.RET) — Example

```
VAH>D PARAM^DSIO0 (.RET) ZW RET
RET="C:\Documents and Settings\AdvDev\Desktop\oCNTFramework\_output\"
```

DSIO OCNT DIALOG LOOKUP

Reference Type

Category OCNT

Description This RPC returns an OCNT Dialog's IEN by classname lookup.

Format DIALK^DSIO0 (.RET, CLASS)

Input Parameters CLASS: The Delphi classname of the dialog form to be opened.

Output SINGLE VALUE

Success A number greater than 0.

Failure 0

DIALK^DSIO0(.RET,CLASS) — Example

```
VAH>D DIALK^DSIO0(.RET,"TdlgBackPain") ZW RET
RET=2
```

```
VAH>D DIALK^DSIO0(.RET,"TdlgBackPainblah") ZW RET
RET=0
```

DSIO OCNT STORE

Reference Type

Category OCNT

Description This RPC will store discreet data for an OCNT or an OCNT dialog.

Format STORE^DSIO0 (.RET, IEN, DATA, DIALOG)

Input Parameters

IEN: The IEN of the entry to the TIU DOCUMENT file (8925).

DATA: A list of values to be saved discreetly and associated to the identified note. Each entry in this list consists of four pieces delimited by caret – TYPE^ID^FIELD^VALUE.

Where TYPE = S, M, WP which means single, multiple, and word processing.

ID is a unique number that associates multiple entries of the list to belong to one entry for VistA.

FIELD is the name of the caption or label that identifies the VALUE which is the fourth piece of a list entry.

S^CONTROL^LABEL 1^SOME VALUE
 S^CONTROL^LABEL 2^SOME OTHER VALUE
 M^CONTROL^M LABEL^VALUE 1^INDEX
 M^CONTROL^M LABEL^VALUE 2^INDEX
 WP^CONTROL^WP LABEL^THIS TYPE IS USED MORE FOR
 A FIELD THAT
 WP^CONTROL^WP LABEL^CONTAINS A LOT OF TEXT
 THAT'S MEANT TO BE READ
 WP^CONTROL^WP LABEL^TOGETHER.

DIALOG: The IEN of the entry to the DSIO OCNT DIALOGS file.

Output

SINGLE VALUE

Success 1

Failure 1

STORE^DSIO0(.RET,IEN,DATA,DIALOG) — Example

```

VAH>ZW DATA
DATA(0)="S^Edit1^Title Edit1:^Edit1 Value"
DATA(1)="S^Edit2^Edit2 Title:^Value of Edit2"
DATA(2)="M^ListBox1^ListBox Header^Value1^0"
DATA(3)="M^ListBox1^ListBox Header^Value2^1"
DATA(4)="WP^Mem01^Title for Mem01:^THIS TYPE IS USED MORE FOR A FIELD THAT"
DATA(5)="WP^Mem01^Title for Mem01:^CONTAINS A LOT OF TEXT THAT'S MEANT TO BE"
DATA(6)="WP^Mem01^Title for Mem01:^"
DATA(7)="WP^Mem01^Title for Mem01:^READ TOGETHER"

VAH>D CREATE^DSIO3(.RET,8,"MCC NOTE",,"TEST DISCREET DATA") ZW RET
RET="5768^1^1"

VAH>D STORE^DSIO0(.RET,5768,.DATA) ZW RET
RET=1

VAH>D ^XUP

Setting up programmer environment
This is a PRODUCTION account.

Terminal Type set to: C-VT100

You have 23 new messages.
Select OPTION NAME: DSIO SHOW DISCREET DATA      DSIO Show Discreet Data for a Note
Select TIU NOTE: MCC NOTE

FIELD: Edit2 Title
SOURCE: OCNT
DATE: APR 21, 2014@13:33:25
CONTROL: Edit2
REMINDER DIALOG:
PARENT:
VALUE:
Value of Edit2

FIELD: ListBox Header
SOURCE: OCNT
DATE: APR 21, 2014@13:33:28
CONTROL: ListBox1
REMINDER DIALOG:
PARENT:
VALUE:
Value1
Value2

FIELD: Title Edit1
SOURCE: OCNT
DATE: APR 21, 2014@13:33:24
CONTROL: Edit1
REMINDER DIALOG:
PARENT:
VALUE:
Edit1 Value

FIELD: Title for Mem01
SOURCE: OCNT
DATE: APR 21, 2014@13:33:28
CONTROL: Mem01
REMINDER DIALOG:
PARENT:

```

VALUE:

THIS TYPE IS USED MORE FOR A FIELD THAT
CONTAINS A LOT OF TEXT THAT'S MEANT TO BE

READ TOGETHER

DSIO OCNT POPULATE

Reference Type**Category** OCNT**Description** This RPC pulls configuration information from VistA by TIU Document Title and OCNT component and returns the values to be initially populated to it.**Format** POP^DSIO0 (.RET, IEN, CONTROL)**Input Parameters** IEN: The IEN of the entry to the TIU DOCUMENT file (8925).

CONTROL: This is the Delphi Component (Object) name.

Output ARRAY

Success

For Multi Valued Components:
-----For TListBox Components with Dialogs:
IEN ^ Display Name ^ ClassOtherwise:
Text ^ IndexFor Single Valued Components:

Text

Failure

Null

POP^DSIO0(.RET,IEN,CONTROL) — Example

```

VAH>D Q^DI

VA FileMan 22.0

Select OPTION: INQUIRE TO FILE ENTRIES

OUTPUT FROM WHAT FILE: DSIO OCNT DIALOGS// DSIO OCNT CONFIGURATION
                        (1 entry)
Select DSIO OCNT CONFIGURATION DOCUMENT DEFINITION: OB H&P INITIAL      TITLE
      Std Title: OB GYN H & P NOTE
ANOTHER ONE:
STANDARD CAPTIONED OUTPUT? Yes//      (Yes)
Include COMPUTED fields:  (N/Y/R/B): NO// - No record number (IEN), no Computed
      Fields

DOCUMENT DEFINITION: OB H&P INITIAL
M CONTROL: LISTBOXCOMPLAINTS
DIALOG: ABDOMINAL PAIN/CRAMPING
DIALOG: BACK PAIN
DIALOG: COUGH/CONGESTION
DIALOG: HEADACHE
DIALOG: LEG PAIN/SWELLING
DIALOG: NAUSEA/VOMITING/DIARRHEA
DIALOG: PALPITATIONS
DIALOG: RASH/ITCHING
DIALOG: VAGINAL BLEEDING
DIALOG: VAGINAL DISCHARGE
DIALOG: VISUAL CHANGES
DIALOG: WHEEZING

VAH>D CREATE^DSIO3(.RET,8,"OB H&P INITIAL",,"TEST OCNT CONFIGURATION") ZW RET
RET="5770^1^1"

VAH>D POP^DSIO0(.RET,5770,"LISTBOXCOMPLAINTS") ZW RET
RET(0)="1^ABDOMINAL PAIN/CRAMPING^TdlgAbdomPain"
RET(1)="10^VAGINAL DISCHARGE^TdlgVagDischarge"
RET(2)="11^VISUAL CHANGES^TdlgVisualChanges"
RET(3)="12^WHEEZING^TdlgWheezing"
RET(4)="2^BACK PAIN^TdlgBackPain"
RET(5)="3^COUGH/CONGESTION^TdlgCoughCongest"
RET(6)="4^HEADACHE^TdlgHeadache"
RET(7)="5^LEG PAIN/SWELLING^TdlgLegPain"
RET(8)="6^NAUSEA/VOMITING/DIARRHEA^TdlgNausea"
RET(9)="7^PALPITATIONS^TdlgPalp"
RET(10)="8^RASH/ITCHING^TdlgRash"
RET(11)="9^VAGINAL BLEEDING^TdlgVagBleed"

```

DSIO OCNT DIALOG POPULATE

Reference Type**Category** OCNT**Description** This RPC preforms the same purpose as DSIO OCNT POPULATE but for OCNT dialogs.**Format** POPD^DSIO0 (.RET, IEN, CONTROL)**Input Parameters** IEN: The IEN of the entry to the DSIO OCNT DIALOGS file.

CONTROL: This is the Delphi Component (Object) name.

Output ARRAY

Success

For Multi Valued Components:

Text ^ Index

For Single Valued Components:

Text

Failure

Null

POPD^DSIO0(.RET,IEN,CONTROL) — Example

```
VAH>D Q^DI

VA FileMan 22.0

Select OPTION: INQUIRE TO FILE ENTRIES

OUTPUT FROM WHAT FILE: DSIO OCNT DIALOGS//
Select DSIO OCNT DIALOGS DISPLAY NAME: VISUAL CHANGES
ANOTHER ONE:
STANDARD CAPTIONED OUTPUT? Yes//      (Yes)
Include COMPUTED fields:  (N/Y/R/B): NO// BOTH Computed Fields and Record Number
(IEN)

NUMBER: 11                                DISPLAY NAME: VISUAL CHANGES
UNIT: udlgVisualChanges                   CLASS: TdlgVisualChanges
M CONTROL: CMBCHAR
INDEX: 0                                ITEM: Blurry
INDEX: 1                                ITEM: Flashing Lights
INDEX: 2                                ITEM: Color
INDEX: 3                                ITEM: Loss of Vision
M CONTROL: CMBLOC
INDEX: 0                                ITEM: Whole Field
INDEX: 1                                ITEM: Segment of Visual Field

VAH>D POPD^DSIO0(.RET,11,"CMBCHAR") ZW RET
RET(0)="Blurry^0"
RET(1)="Flashing Lights^1"
RET(2)="Color^2"
RET(3)="Loss of Vision^3"
```

DSIO OCNT RESTORE

Reference Type

Category OCNT

Description This RPC behaves like the population RPCs but instead it populates OCNT components with discreet data saved by TIU Document Title.

Format RESTORE^DSIO0(.RET,IEN,DIALOG)

Input Parameters IEN: The IEN of the entry to the TIU DOCUMENT file (8925).

DIALOG: The IEN of the entry to the DSIO OCNT DIALOGS file.

Output ARRAY

Success Multiple Line Items of: Component Name ^ Value

Failure Null

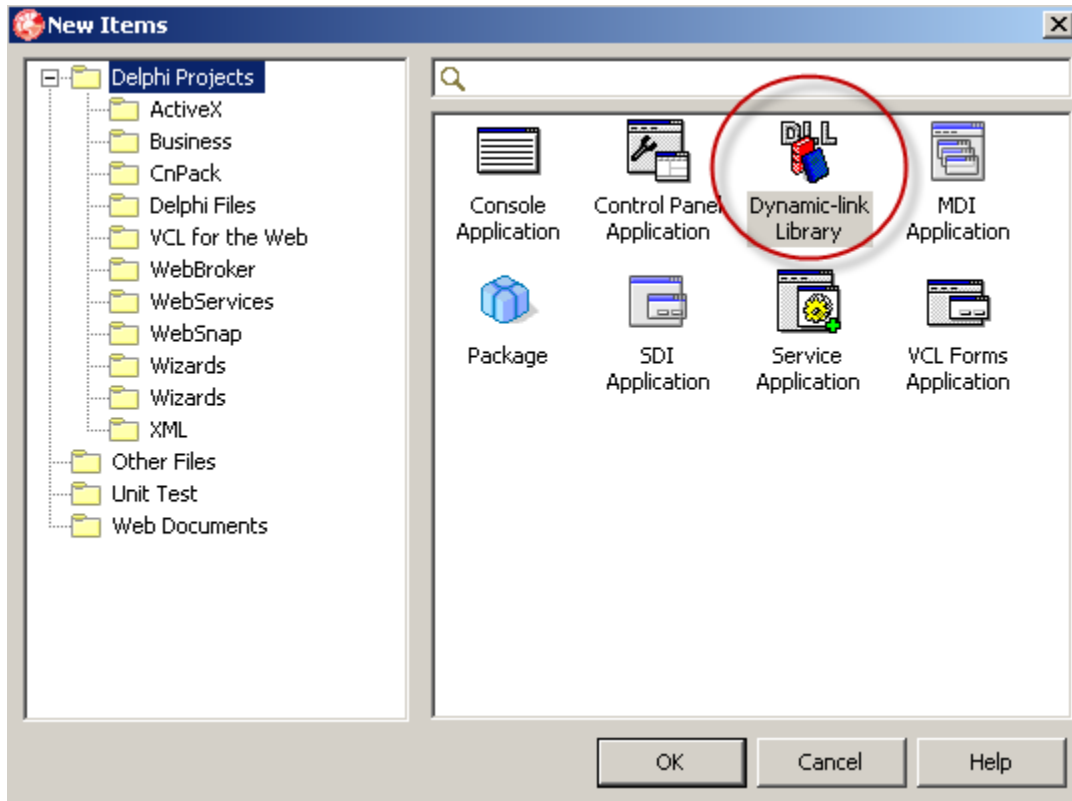
RESTORE^DSIO0(.RET,IEN,DIALOG) — Example

```
VAH>D RESTORE^DSIO0(.RET,5770) ZW RET
RET(0)="Edit1^Edit1 Value"
RET(1)="Edit2^Value of Edit2"
RET(2)="ListBox1^Value1^0"
RET(3)="ListBox1^Value2^1"
RET(4)="Memo1^THIS TYPE IS USED MORE FOR A FIELD THAT"
RET(5)="Memo1^CONTAINS A LOT OF TEXT THAT'S MEANT TO BE"
RET(6)="Memo1^"
RET(7)="Memo1^READ TOGETHER"
```

Development

*Delphi Development Examples done in D2010.

Creating an OCNT is like starting any other DLL or “Dynamic-link Library” project.



Now, there are a couple of requirements that we have to meet in order to allow the COMObject to control our OCNT and another to use the broker.

So, in order to give control to the COMObject we need to add a function to our DLL source. First add a new form to the project then add the circled code (make sure you add ShareMem and Forms to your uses clause):

```
Library Project1;

{ Important note about DLL memory management: ShareMem must be the
first unit in your library's USES clause AND your project's (select
Project-View Source) USES clause if your DLL exports any procedures or
functions that pass strings as parameters or function results. This
applies to all strings passed to and from your DLL--even those that
are nested in records and classes. ShareMem is the interface unit to
the BORLNDMM.DLL shared memory manager, which must be deployed along
with your DLL. To avoid using BORLNDMM.DLL, pass string information
using PChar or ShortString parameters. }

uses
  ShareMem,
  SysUtils,
  Classes,
  Forms,
  Unit1 in 'Unit1.pas' {Form1};

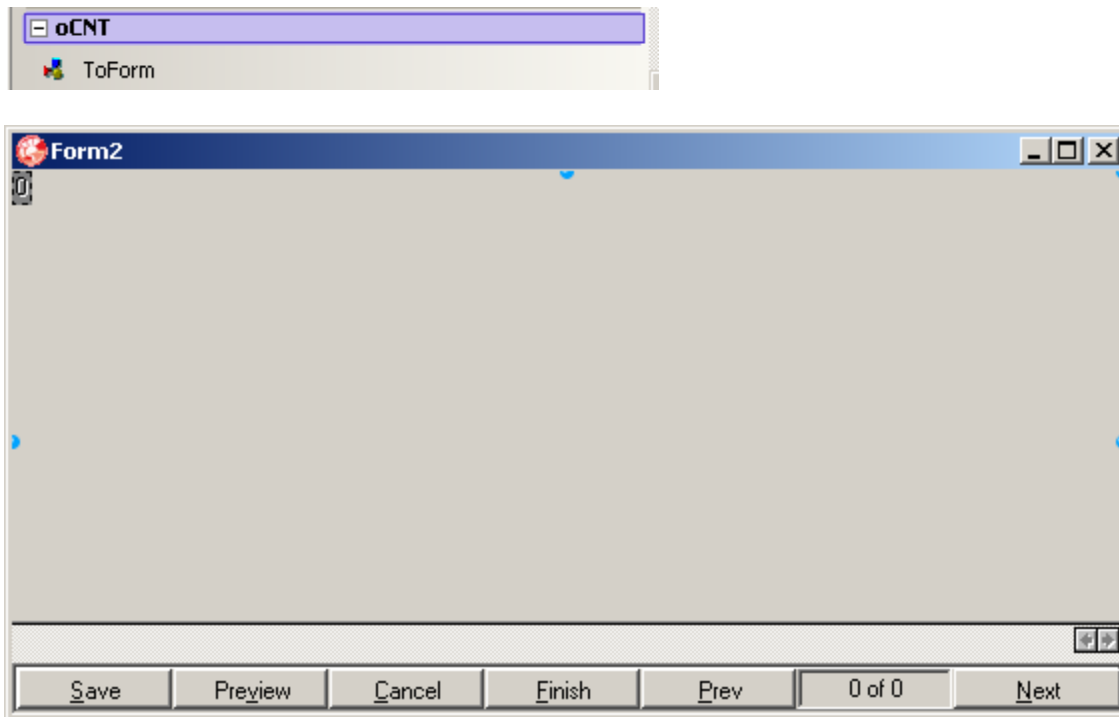
{$R *.res}

function LaunchCNT: Pointer; stdcall;
begin
  Form1 := TForm1.Create(Application);
  Result := @Form1;
end;

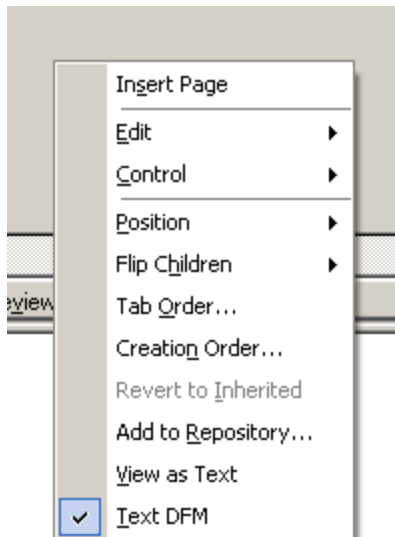
exports
  LaunchCNT;

begin
end.
```

Next, you your tool palette search for the OCNT category, it should only have a component called ToForm, with it, add it to your unit. **The ToForm should always be placed directly on a TForm.**



The ToForm component is named ofrm1 and the name is a read only property and this is by design as only one of these components are allowed for any given project.

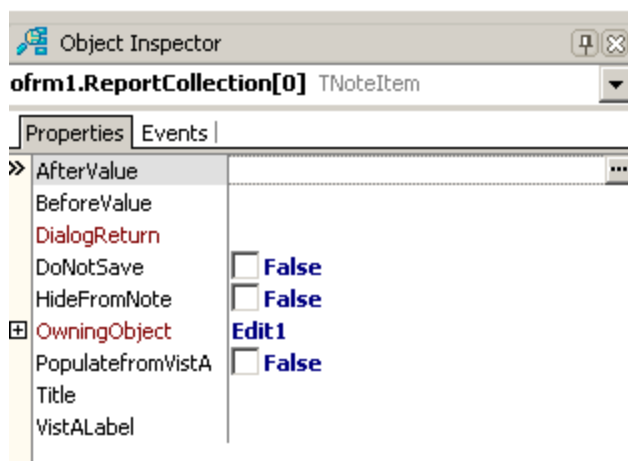
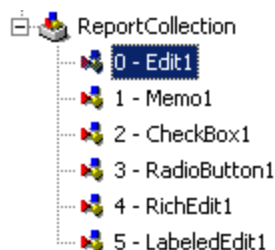
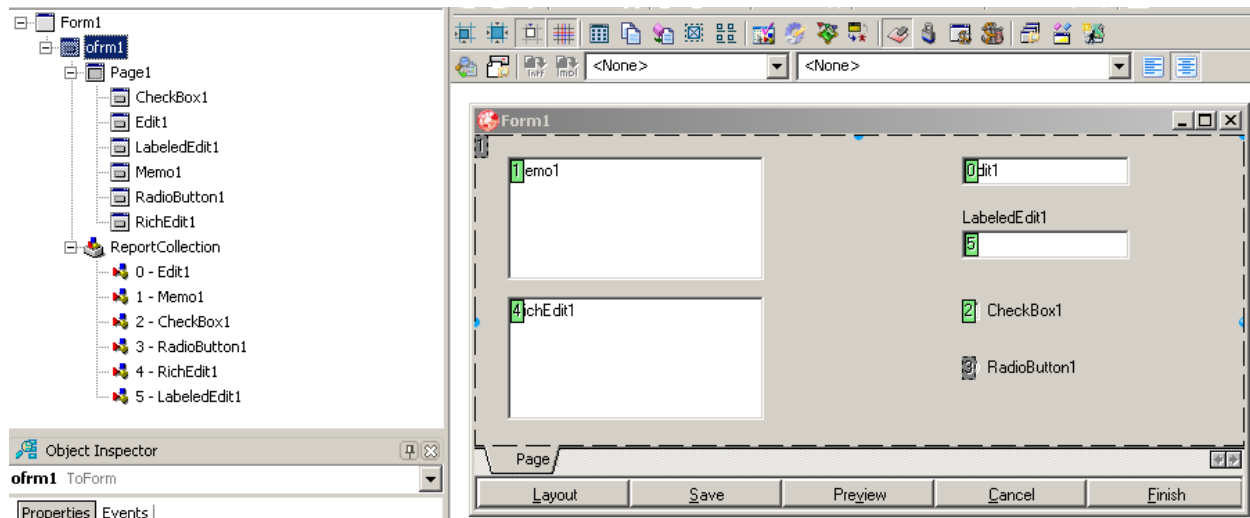


The first thing you'll want to do is to add a page. You don't want to start throwing controls on the form as it is now. Only controls placed on a page can be added to a note.

The image shows a screenshot of a software window titled "Form2". The window has a standard Windows-style title bar with a red icon on the left and minimize, maximize, and close buttons on the right. The main area of the window is a large, empty, light gray rectangle, representing a form. Below this area is a navigation bar. On the left of the navigation bar is a tab labeled "Page1". To the right of the tab are two small arrows. Below the navigation bar is a row of buttons: "Save", "Preview", "Cancel", "Finish", "Prev", "0 of 1", and "Next".

ReportCollection

Now, when you add a control to a page it will be added to your ReportCollection, with the ReportCollection you'll be able to extend any Window's based control to interact with VistA and be added to our note. Later updates of the ToForm component could include additional controls.



Each of the report items are connected to a control (don't change the **owningobject** this was left as a published property so you can identify what control the item belongs to. In the case of a TListBox it can also have a **DialogReturn** object as a TMemo or TRichEdit so if it has been set to **PopulatefromVistA** True and say it is named ListBox1. If it has a DSIO OCNT CONFIGURATION entry of LISTBOX1 and it is has dialog entries then when ran it will populate itself with that list allowing it to open Dialogs with the oCNTDialog.dll.

VistALabel is for saving this control's value(s) to VistA discreetly. If you want to save it to VistA then this property must be set.

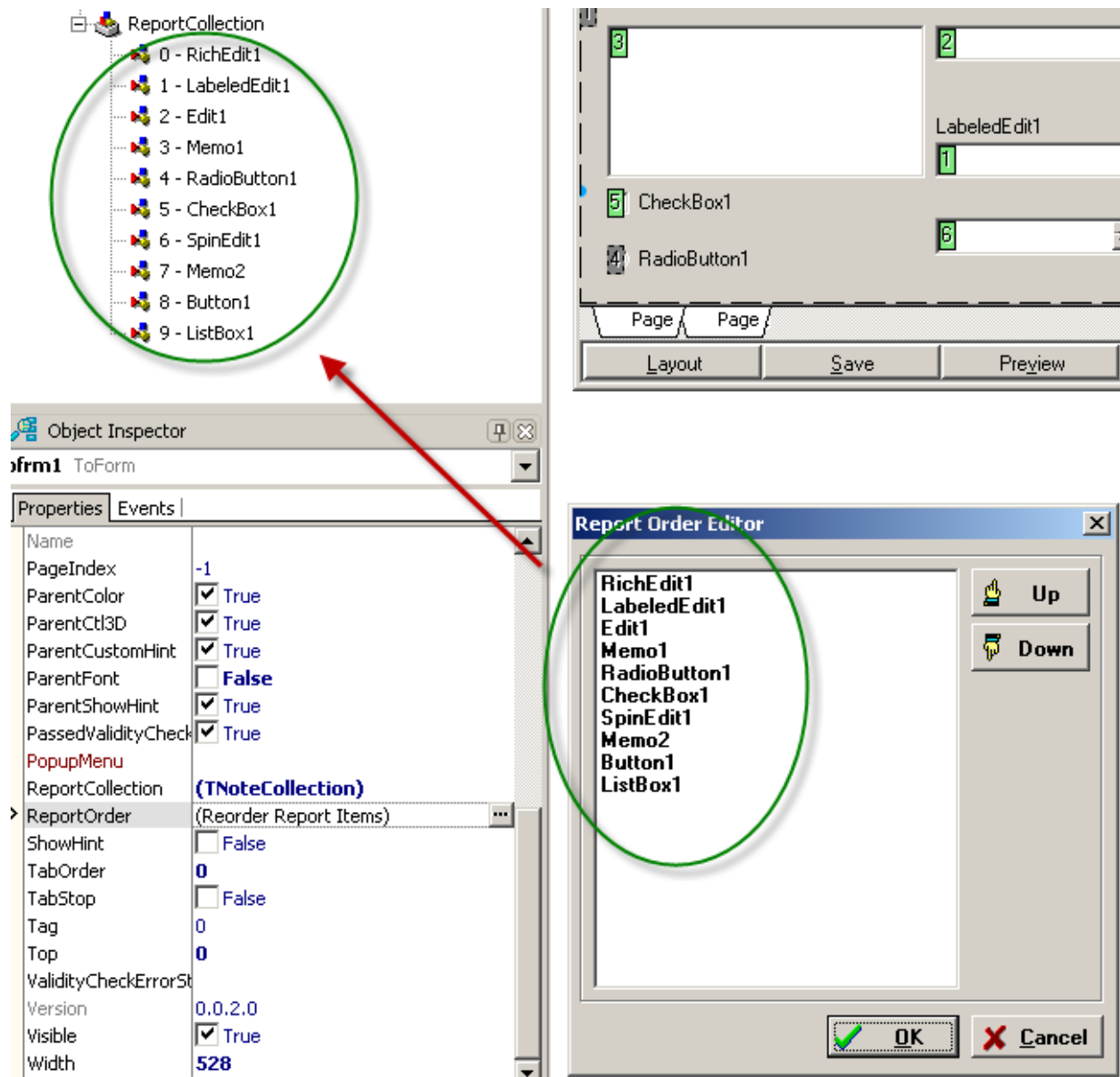
A **BeforeValue** and **AfterValue** are note labels for the control's value such as inches for an aftervalue to a control capturing inches. **Title** acts the same way but is populated to the note on the line before the value is displayed.

If a value isn't to be saved to VistA discreetly then the **DoNotSave** property needs to be set to True, while setting the **HideFromNote** property to True prevents the control from being populated to the generated note.

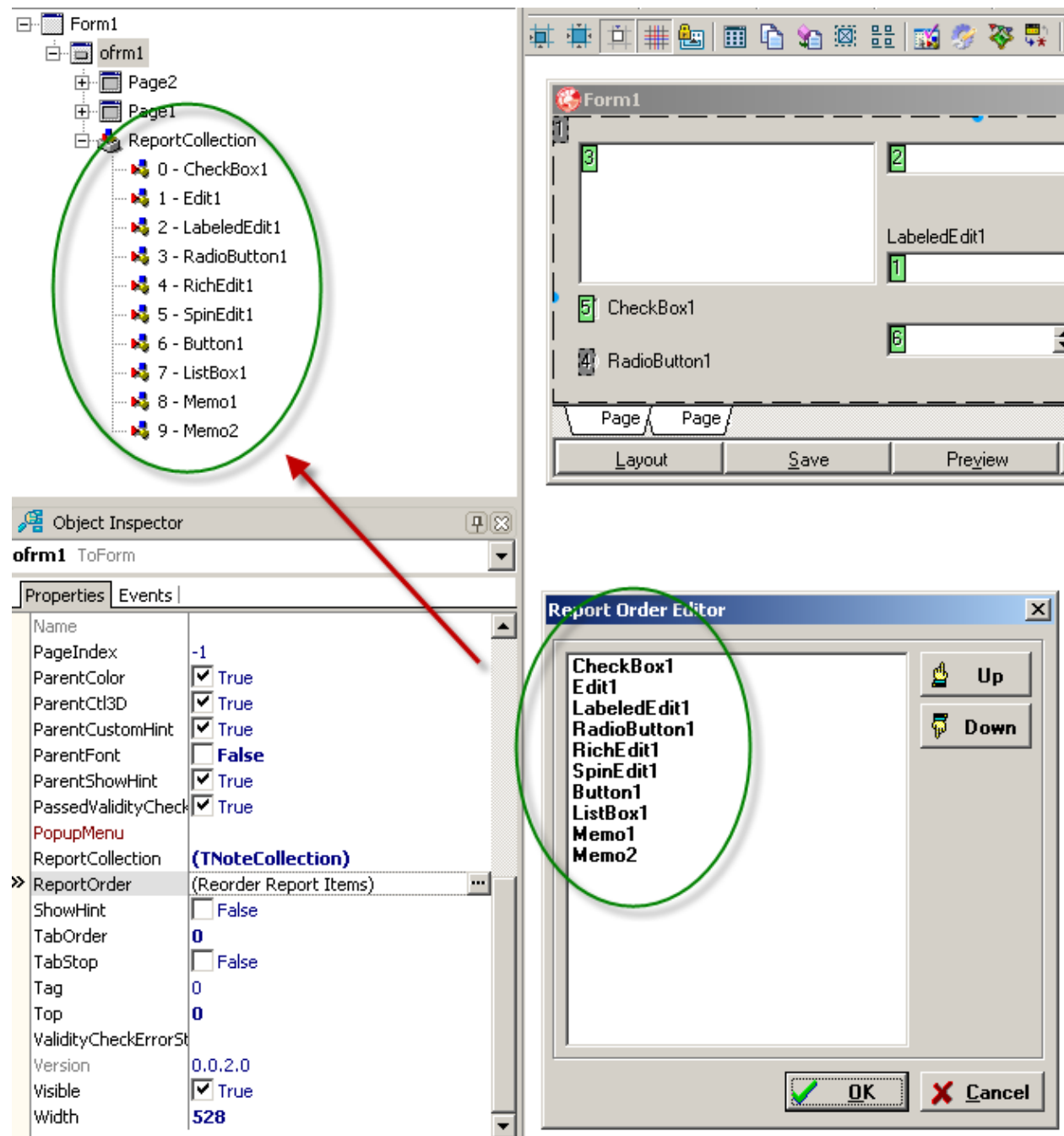
ReOrder ReportCollection Items:

The ReportCollection also determines the order of which elements are added to the note.

Before:



After:



Title, BeforeValue, and AfterValue

ofrm1.ReportCollection[0] TNoteItem		ofrm1.ReportCollection[1] TNoteItem	
Properties	Events	Properties	Events
AfterValue	lbs	AfterValue	
BeforeValue	Patient's Weight:	BeforeValue	
DialogReturn		DialogReturn	
DoNotSave	<input type="checkbox"/> False	DoNotSave	<input type="checkbox"/> False
HideFromNote	<input type="checkbox"/> False	HideFromNote	<input type="checkbox"/> False
OwningObject	Edit1	OwningObject	Memo1
PopulatefromVistA	<input type="checkbox"/> False	PopulatefromVistA	<input type="checkbox"/> False
Title		Title	Dialog Return: []
VistALabel	WEIGHT	VistALabel	

Review Note

Patient's Weight: 200 lbs

Dialog Return:

Abdominal pain/cramping:

Nausea/vomiting/diarrhea? Yes

Appetite? Yes

Fever? Yes

Accept Note

Return To Form

PopulatefromVistA

VAH>D Q^DI

VA FileMan 22.0

Select OPTION: ENTER OR EDIT FILE ENTRIES

INPUT TO WHAT FILE: DSIO OCNT

CONFIGURATION//

EDIT WHICH FIELD: ALL//

Select DSIO OCNT CONFIGURATION DOCUMENT

DEFINITION: MCC NOTE

DOCUMENT DEFINITION: MCC NOTE//

Select CONTROL: LISTBOX1//

CONTROL: LISTBOX1//

Select DIALOGS: WHEEZING// ?

Answer with DIALOGS

Do you want the entire 12-Entry DIALOGS List? Y (Yes)

Choose from:

ABDOMINAL PAIN/CRAMPING

BACK PAIN

ofrm1.ReportCollection[6] TNoteItem	
Properties	Events
AfterValue	
BeforeValue	
DialogReturn	Memo1
DoNotSave	<input checked="" type="checkbox"/> True
HideFromNote	<input type="checkbox"/> False
OwningObject	ListBox1
PopulatefromVistA	<input checked="" type="checkbox"/> True
Title	
VistALabel	

COUGH/CONGESTION
HEADACHE
LEG PAIN/SWELLING
NAUSEA/VOMITING/DIARRHEA
PALPITATIONS
RASH/ITCHING
VAGINAL BLEEDING
VAGINAL DISCHARGE
VISUAL CHANGES
WHEEZING

The screenshot shows a software form titled "Form1" with a blue title bar. The form contains several controls: a "Memo1" text area, a "RichEdit1" text area, a "CheckBox1" with an unchecked box, a "RadioButton1" with an unchecked circle, an "Edit1" text box, and a "LabeledEdit1" text box. A list box is positioned in the center-right, containing the following items: "ABDOMINAL PAIN/CRAMPING", "VAGINAL DISCHARGE", "VISUAL CHANGES", "WHEEZING", "BACK PAIN", "COUGH/CONGESTION", and "HEADACHE". This list box is circled in red. At the bottom of the form, there is a "Page1" tab and a row of buttons: "Save", "Preview", "Cancel", "Finish", "Prev", "1 of 1", and "Next".

The screenshot shows a software window titled "Form1". Inside, there are several input fields: a large empty text box, a "RichEdit1" box, a "CheckBox1", a "RadioButton1", and two "Edit1" text boxes. A list box on the right contains "ABDOMINAL PAIN/CRAMPING", "VAGINAL DISCHARGE", and "VISUAL CHANGES". A modal dialog box titled "Abdominal pain/cramping" is open in the foreground. The dialog has a title bar with a close button. Below the title, it says "Enter appropriate information. Press OK to close." The dialog contains several questions with checkboxes: "Onset:" followed by a text box; "Contractions?" with "Yes" and "No" checkboxes; "Nausea/vomiting/diarrhea?" with "Yes" and "No" checkboxes; "Appetite?" with "Yes" and "No" checkboxes; "Fever/chills?" with "Yes" and "No" checkboxes; and "Location:" followed by a text box. At the bottom of the dialog are "OK" and "Cancel" buttons.

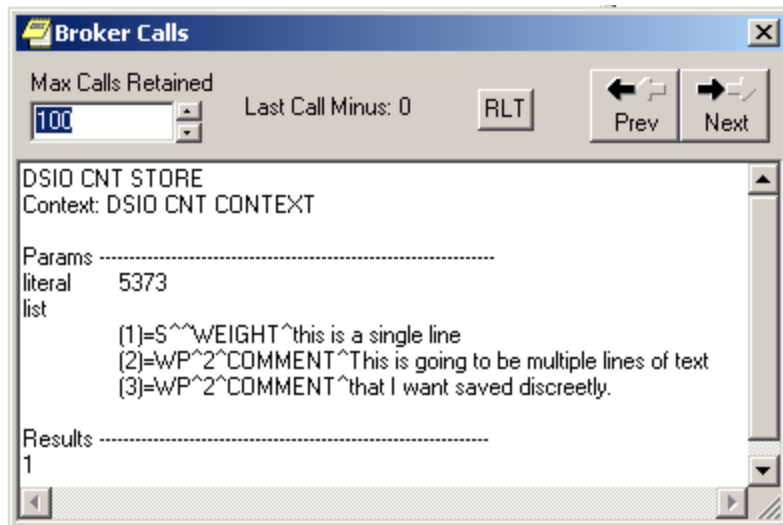
*Multiple dialogs can populate a single TMemor or TRichEdit it doesn't delete any text before adding.

This screenshot shows the same "Form1" window, but the modal dialog is closed. The text from the dialog is now entered into the "RichEdit1" text box. The text is: "Abdominal pain/cramping:", "Nausea/vomiting/diarrhea? Yes", "Appetite? Yes", and "Fever? Yes". A red circle is drawn around this text in the "RichEdit1" box. The list box on the right now shows a longer list: "ABDOMINAL PAIN/CRAMPING", "VAGINAL DISCHARGE", "VISUAL CHANGES", "WHEEZING", "BACK PAIN", "COUGH/CONGESTION", and "HEADACHE". At the bottom of the window, there is a "Page1" tab and a series of buttons: "Save", "Preview", "Cancel", "Finish", "Prev", "1 of 1", and "Next".

ofrm1.ReportCollection[1] TNoteItem		ofrm1.ReportCollection[0] TNoteItem	
Properties Events		Properties Events	
>> AfterValue	...	>> AfterValue	lbs
BeforeValue		BeforeValue	Patient's Weight:
DialogReturn		DialogReturn	
DoNotSave	<input type="checkbox"/> False	DoNotSave	<input type="checkbox"/> False
HideFromNote	<input type="checkbox"/> False	HideFromNote	<input type="checkbox"/> False
⊕ OwningObject	Memo1	⊕ OwningObject	Edit1
PopulatefromVistA	<input type="checkbox"/> False	PopulatefromVistA	<input type="checkbox"/> False
Title	Dialog Return:	Title	
VistALabel	COMMENT	VistALabel	WEIGHT

Saving Discreet Data

The screenshot shows a report form titled "Form1". It contains two text input fields. The first field is a multi-line text area containing the text "This is going to be multiple lines of text that I want saved discreetly." The second field is a single-line text box containing the text "this is a single line". At the bottom of the form, there is a navigation bar with buttons labeled "Save", "Preview", "Cancel", "Finish", "Prev", "1 of 1", and "Next".



VAH>D Q^DI

VA FileMan 22.0

Select OPTION: 1 ENTER OR EDIT FILE ENTRIES

INPUT TO WHAT FILE: DSIO OCNT CONFIGURATION// DSIO NOTE DATA
(81 entries)

EDIT WHICH FIELD: ALL//

Select DSIO NOTE DATA: `5373 MCC NOTE

NOTE: MCC NOTE//

Select DATA: COMMENT// ?

Answer with DATA

Choose from:

COMMENT

WEIGHT

You may enter a new DATA, if you wish

Enter one of the following:

M.EntryName to select a MULTIPLE

WP.EntryName to select a WORD PROCESSING

S.EntryName to select a SINGLE

To see the entries in any particular file type <Prefix.??>

VAH>ZW ^DSIO(19641.41,5373)

^DSIO(19641.41,5373,0)=5373

^DSIO(19641.41,5373,1,0)="^19641.411AV^2^2"

^DSIO(19641.41,5373,1,1,0)="51;DSIO(19641.43,"

^DSIO(19641.41,5373,1,2,0)="17;DSIO(19641.45,"

^DSIO(19641.41,5373,1,"B","17;DSIO(19641.45","2)=""
^DSIO(19641.41,5373,1,"B","51;DSIO(19641.43","1)=""

VAH>ZW ^DSIO(19641.43,51)
^DSIO(19641.43,51,0)="WEIGHT"
^DSIO(19641.43,51,1)="this is a single line"
^DSIO(19641.43,51,2)="5373^OCNT^3140312.110527"

VAH>ZW ^DSIO(19641.45,17)
^DSIO(19641.45,17,0)="COMMENT"
^DSIO(19641.45,17,1,0)="^^2^2^3140312"
^DSIO(19641.45,17,1,1,0)="This is going to be multiple lines of text"
^DSIO(19641.45,17,1,2,0)="that I want saved discreetly."
^DSIO(19641.45,17,2)="5373^OCNT^3140312.110527"

Additional Reading

For additional reading please check out the VA's Software Document Library:

<http://www.va.gov/vdl/>

For CPRS try: <http://www.va.gov/vdl/application.asp?appid=61>

The CPRS Technical Manual: GUI Version (Appendix F) additional information from a development and configuration standpoint for working with COMObjects.