Provide Enterprise

BROUGHT TO YOU BY
GROUPWARE TECHNOLOGIES

Reporting Basics

Provide Enterprise User Guide

Version 5.1

© 2024 Groupware Technologies, Inc. All rights reserved. The GTI logo and Provide® Enterprise are registered trademarks of Groupware Technologies, LLC. All other product and company names referenced herein may be trademarks or registered trademarks of their respective companies.

Last Updated June 26, 2024

Table of Contents

Chapter One: Introduction 1

Chapter Two: Understanding Document Storage 2

Definition of a Document 2

Document Storage 2

Document Tables 2

tblDocument_Info 2

Data Tables tblform_name_[field_name] 3

tblProvide_Document_Summary 3

tblDocument_Access 4

Bucket Tables 4

Walk through of Saving a Document 5

Step 1 5

Step 2 5

Step 3 5

Step 4 5

Step 5 5

Walkthrough of Viewing a Document 6

As Of Date 6

Document Versions 6

Chapter Three: Understanding Table Relationships 8

As we learned in the previous chapter, documents can be stored in many tables. In this chapter we will cover how the table relationships define the document. 8

Linking Tables from Different Documents by Client 9

Linking Tables from an Activity Document to a Reference Document 10

Chapter 4: Creating Crystal Reports for Provide® Enterprise 11

Using Parameter Fields and Selection Criteria in Embedded Crystal Report Objects 15

Client Parameter Fields and Selection Criteria 16
Creating a Parameter Field 16
Adding a Selection Criteria to the Report 16
Suggested Parameter Field Prompts and Selection Criteria – Client Reports 17
Parameter Field Prompts 17
Selection Criteria Formula 18
Suggested Parameter Field Prompts and Selection Criteria – Document Reports 18
Parameter Field Prompts 18
Selection Criteria Formula 18
Chapter 5: Managing Crystal Report Objects in Provide® Enterprise 19
Adding a New Report Object 19
Deleting a Report Object 21

Chapter One: Introduction

While most users of the Provide® Enterprise system use it to track, manage and coordinate the care of Clients, there are always those people within an organization responsible for reporting on and accounting for the activities of the staff that use Provide® Enterprise every day. Reporting in Provide® Enterprise is limited only by the ability of agencies to create their own reports and/or link to the data in the Provide® Enterprise tables using various reporting applications.

Chapter Two: Understanding Document Storage

This chapter will review how document storage functions in Provide Enterprise.

Definition of a Document

Within Provide® Enterprise, a document is a set of related information that records a unit of knowledge or work. The document's information may be formatted on a standard form, just as a paper form can be used to document information. While the form may include headers, footers and static text, the document actually contains the information that is collected. Characteristics of a document include:

- Authoring information - who edited it and when.
- Access control - who may see it or change it.
- Life span - a creation date, modified dates, various state transitions and modifications, destruction. Important to this concept is that a document's existence has a lasting impact even after it is deleted. The fact that a document is deleted does not nullify the impact that it had. For this reason, even deleted documents are retained internally for auditing purposes.
- Named fields - information is stored in named fields. These fields can be of different types of data (text, numbers, dates, rich text) and differing quantities (single or multiple values).

This definition is the basis for the design of the database schema as described in the next section.

Document Storage

Because each implementation of Provide® Enterprise may have a different database schema depending on customer requirements, Provide® Enterprise uses a model architecture to define how the schema is to be built. Key to the architecture is the support of document storage.

Document Tables

tblDocument_Info

This is the base document table, which contains the following fields:

Form	The name of the form as specified in the Data Dictionary SQLForm field.
DocID	The document unique ID throughout the document lifespan.
SQLID	The document unique ID for a given point in time in which the document was saved. There may be multiple SQLID's for a given DocID.
ModDate	The GMT timestamp of when the document was saved according to the clock of the system where it was saved.
LocalModDate	The GMT timestamp of when the document was saved in this database.
RepID	The system ID where the document was updated. This is the Provide® Enterprise System ID as found in the Publish Control record.
UName	The user ID that updated the document.
ALimitAccess	Limit access flag. If this flag is set to "Yes", then regular security is bypassed and the document reader list must be manually set.
DeleteFlag	Y or N. Marks whether or not this document is deleted in Provide® Enterprise.

Data Tables tblform_name_[field_name]

Single-value fields may be combined into common tables. Multiple value fields should be isolated to their own tables. Table names begin with "tbl", which is prefixed to the SQL Table name defined in the Data Dictionary Field. The suggested table naming convention is to use the SQL form name and replace all special characters with underscores. Multiple value tables should then suffix the form name with the field name. A document may have several data tables to store all the data related to that document type. User access to the tables is restricted to insert new records only.

SQLID	Same as in tblDocument_Info
IDX	The zero-based index of the value, for multi-valued fields. Defaults to 0.
Field…	The data field columns in alphabetical order

tblProvide_Document_Summary

This table contains the SummaryText field for all Client, Activity and Reporting documents. It also contains the necessary information so that access control lists can be calculated after the record is inserted.

SQLID	Same as in tblDocument_Info
IDX	The zero-based index of the value, for multi-valued fields. Defaults to 0.

ClientProfileID	The internal unique Client ID
Form	The SQL Form Name
SubForm	For Assessments, the type of assessment
Aorg	Organization creating the form
Apgm	Program creating the document
AactivityDate	Activity Creation Date
SummaryText	The Selector or SelectorDescription

tblDocument_Access

This table lists the read and write groups for each SQLID instance of a document. It is only used for Client, Activity and Reporting documents.

Form	The SQL form name
AOrg	The organization
APgm	The program
DocID	The document unique id
SQLID	The instance SQLID
AccessDate	The effective access date
AccessType	"R" for read, "W" for write
AppliedDate	The date this record was created

Bucket Tables

Bucket tables are used to store information that is not deemed important to report against, but are necessary for the Provide® Enterprise application. They are also used to store fields not defined in the Data Dictionary. Bucket tables include tblchar_bucket, tblvarchar_bucket, tbldatetime_bucket, tblint_bucket, and tblfloat_bucket. The format of these tables is: Character fields are limited to 7500 characters. Text fields are limited to 2GB.

SQLID	Same as in tblDocument_Info
IDX	The zero-based index of the value, for multi-valued fields. Defaults to 0.
FieldName	The name of the document field
BucketValue	The value of the document field

Walk through of Saving a Document

The process of saving a document involves many steps that are bunched into a transaction. By bunching these steps into a transaction, the system will not store partially processed documents.

Step 1

Establish the start of a transaction.

Step 2

sp_Provide_Update_Document is called. This stored procedure first validates the data needed to be written into tblDocument_Info. It then checks to see if the User saving the document has the access to do so by calling fn_Validate_Provide_Document_Action. This function then takes the document type and checks it against the User's permissions/roles. If everything checks out, a record is inserted into tblDocument_Info for the current document. If not, the transaction is rolled back (all information previously written to the database in this transaction is removed) and an error is returned stating why the record could not be saved (Incorrect data sent or user does not have permission).

Step 3

Information about what data tables this document is stored in is read from the data dictionary. Then for each table defined in the dictionary the corresponding fields are read from the document and saved in each table. Each data table has a trigger that is run just prior to the record being written to the database that checks that there is a record in tblDocument_Info that matches. If no matching record is found, the transaction ID is rolled back and an error is returned.

Step 4

For documents that require further security, such as Client information, Client activity and reporting, a record is added to the tblProvide_Document_Summary. This table contains a trigger that calls sp_CalculateSecurityForDocument. This stored procedure will add records to tblDocument_Access for the organization – programs that should have access to this document.

Step 5

If no errors were encountered, then the transaction is completed and ended. The document is now saved in the database.

Walkthrough of Viewing a Document

The viewing of documents or parts of a document is controlled by the use of views. Two views exist for every data table that is defined in the database: the secure view and the secure data view.
User access to the secure view is not allowed and is restricted to read only for the secure data view. The secure view creates a list of documents that the current User has access to see. This is done by taking the organization and program stored in the User's current user profile, the As Of Date (see below) and joining it with records that match the document type, organization, program and date modified in tblDocument_Access. The secure data view then joins the secure view with the data table to get all the data in the data table that the User has access to see. A stored procedure for each document type is used by the system to easily load all the data from all the tables that are for a given document type.

As Of Date

Another function of the view is to display only the most recent version of the document as it was on the As Of Date. By default, the As Of Date is set to the future and the document returned is the most recent document saved. This date can be changed in the current user profile to reflect a date in the past. This will cause the document to be displayed as it was on the date set in the current user profile.

Document Versions

Each time a change is made to a document, new records are written to the tables (see Saving a Document). This creates multiple "versions" of each document. Each version of the document is assigned a SQLID that is a unique identifier.

Table has following fields:

SQLID	DocID	Date Modified
123	456	1/1/2000
124	456	1/2/2000

125	456	1/3/2000

The view would show:

SQLID	DocID	Date Modified
125	456	1/3/2000

Chapter Three: Understanding Table Relationships

As we learned in the previous chapter, documents can be stored in many tables. In this chapter we will cover how the table relationships define the document.

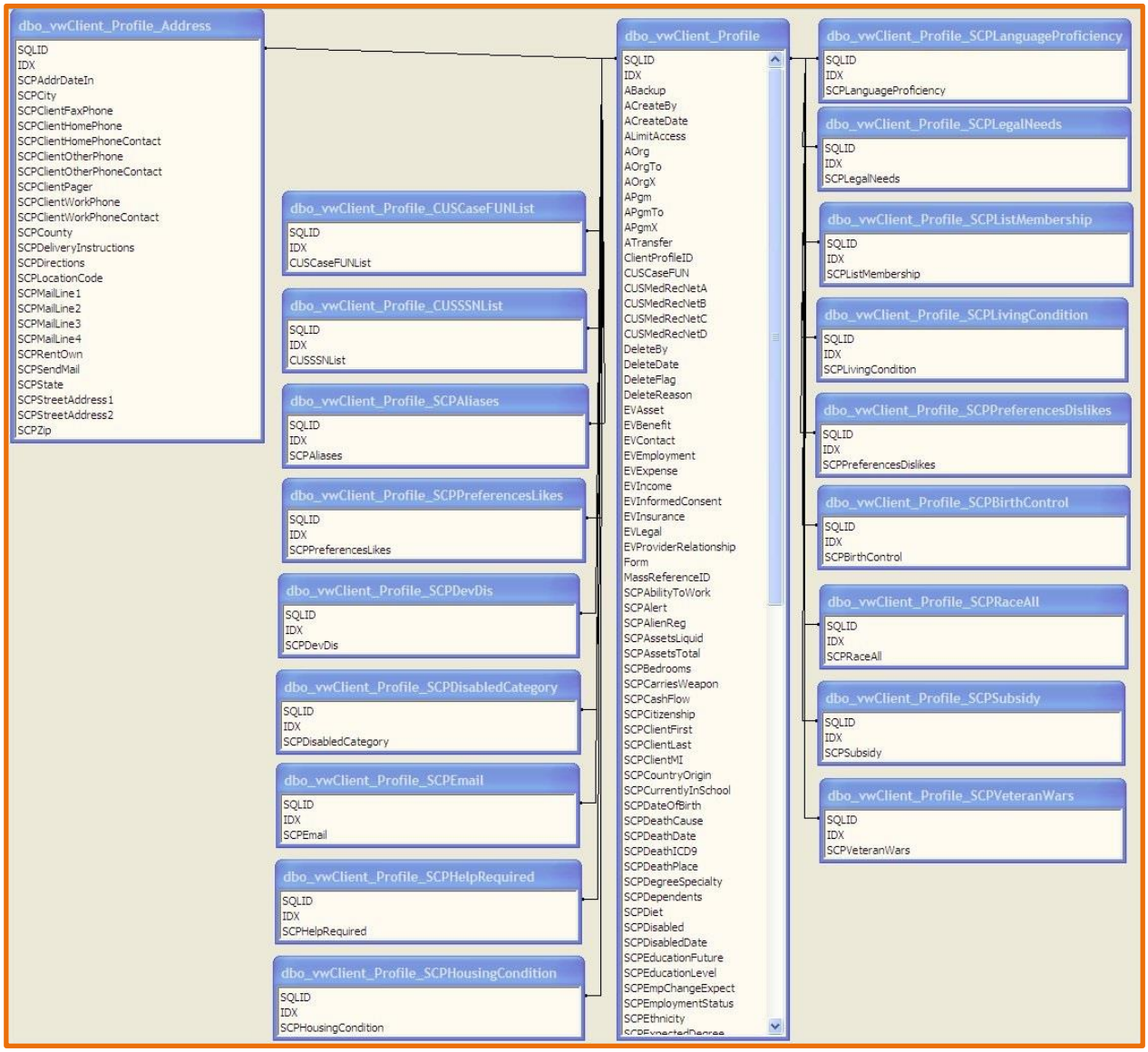


Figure 1

In Figure 1 we show the links for the Client Profile Document. You can see all the multi-value fields are linked to the Client Profile Table by SQLID. These tables can contain

several records with the same SQLID. The IDX field in these tables along with the SQLID provides a unique key for the table.

Linking Tables from Different Documents by Client

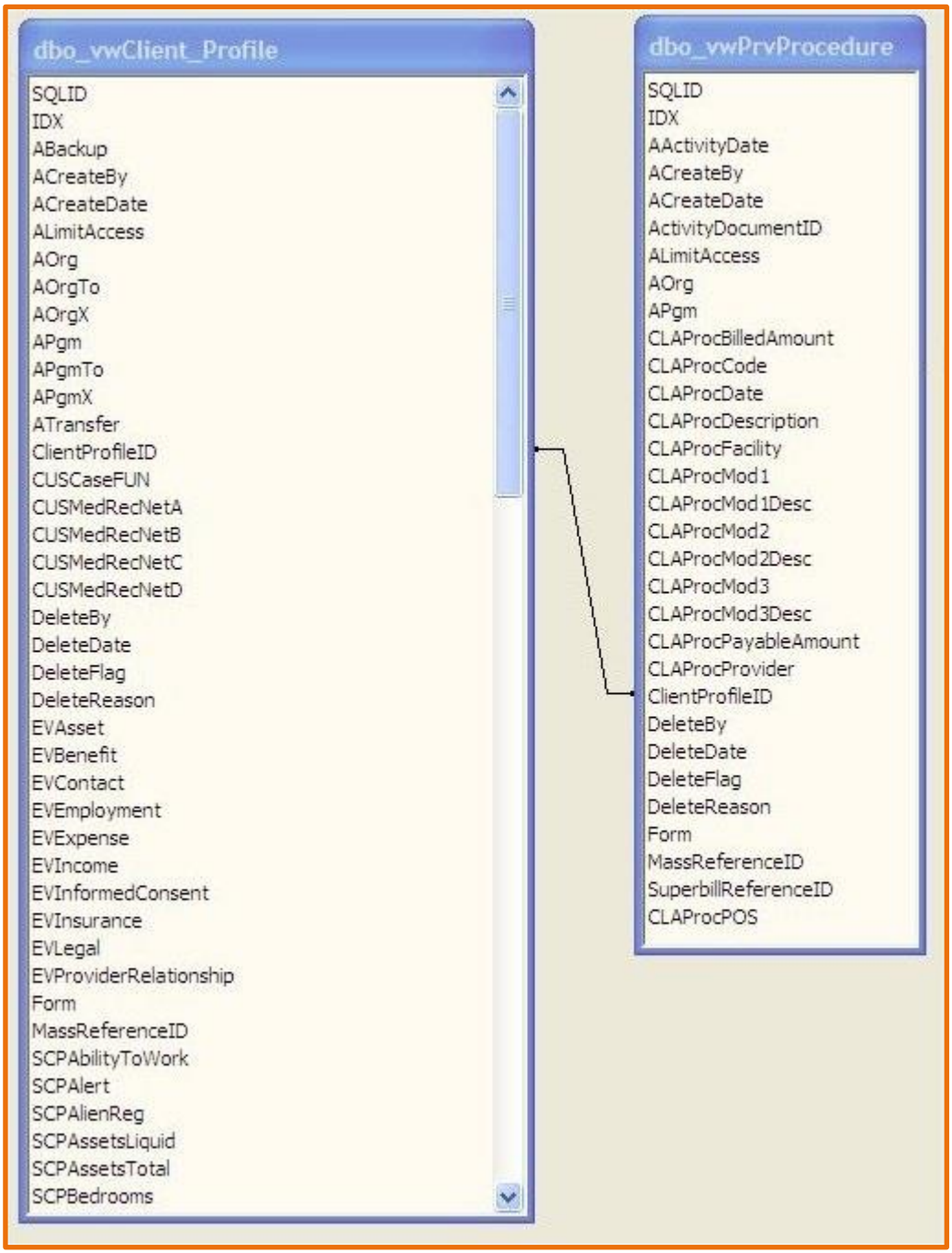


Figure 2

In figure 2 we show the link of the Client Profile to a Procedure. You can link Client records by ClientProfileID on any activity documents main document table. You can then also link sub tables for the various documents to the documents main table.

Linking Tables from an Activity Document to a Reference Document

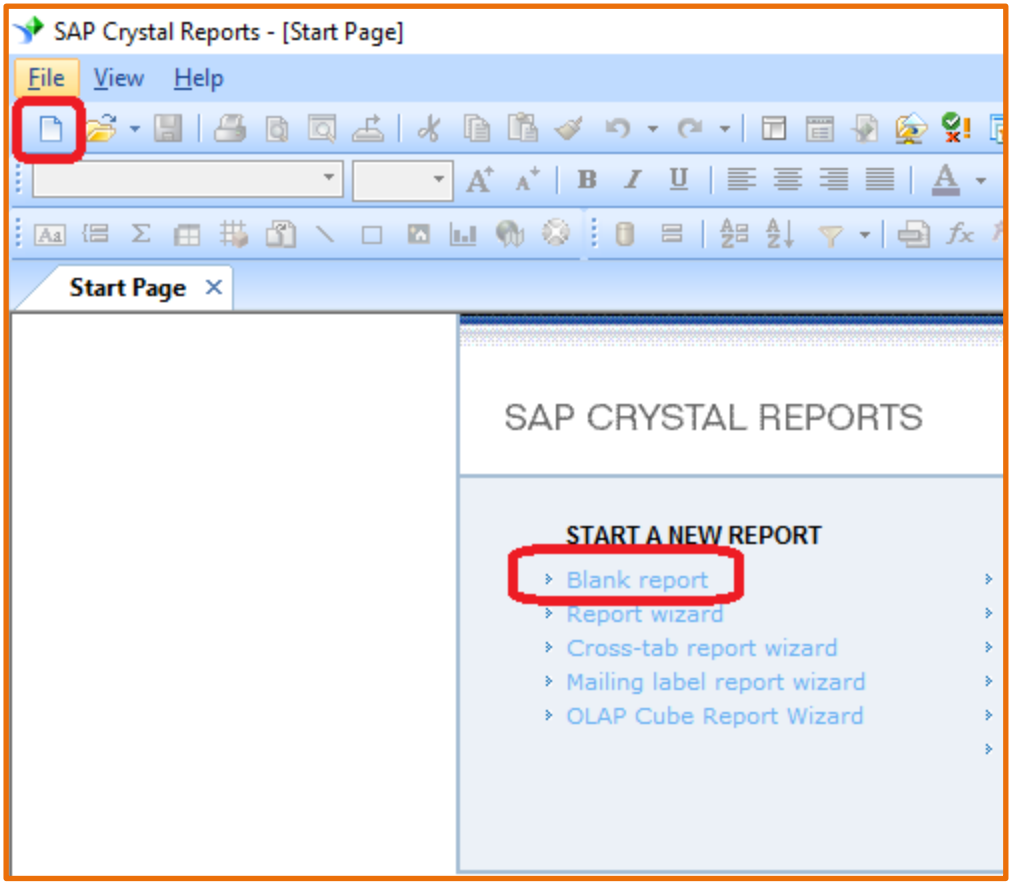
Figure 3

In figure 3 we show the link of the Procedure to the CPT Procedure Codes. You can link Client procedure records to the CPT Procedure Reference Table by the Procedure Code. You can then also link sub tables for the various documents to the documents main table of the procedure or the CPT Procedure Code table.

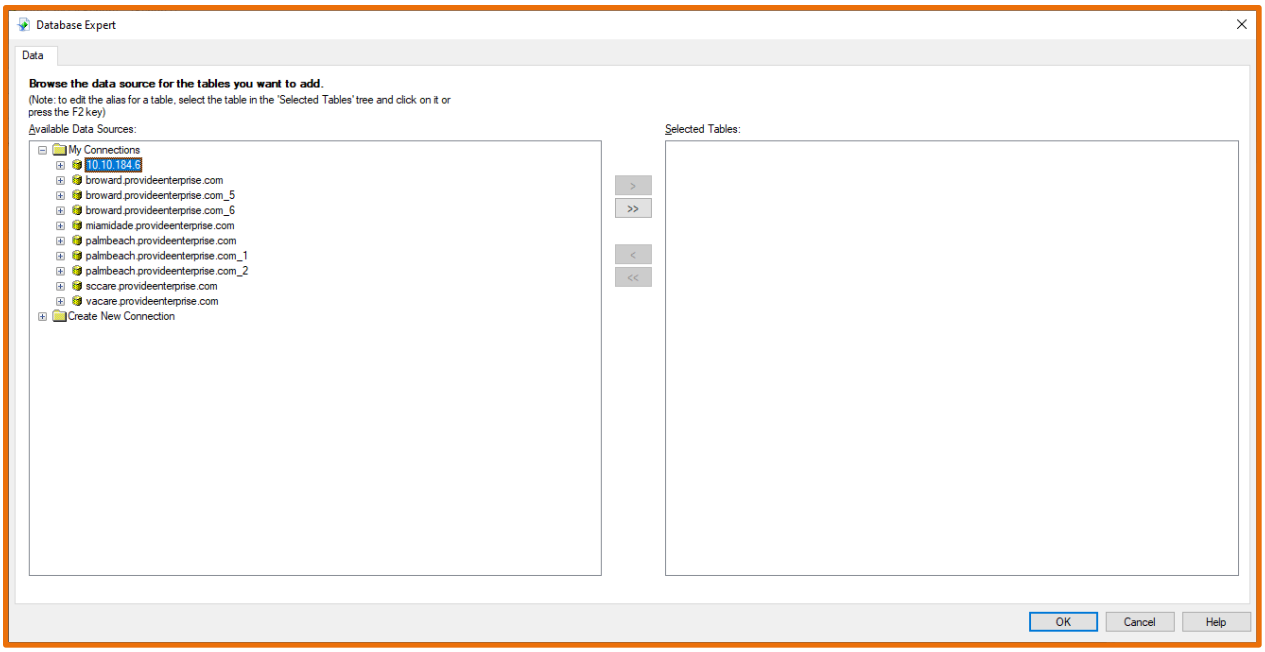
Chapter 4: Creating Crystal Reports for Provide® Enterprise

Crystal Reports version 8.5 and 11.5 are excellent tools for creating reports that can be run directly from Provide® Enterprise. Outlined below are the steps needed to design embedded Crystal Reports effectively for use within Provide® Enterprise. This chapter is not meant to be an overview of Crystal Reports and assumes that the user already is familiar with this report building tool.

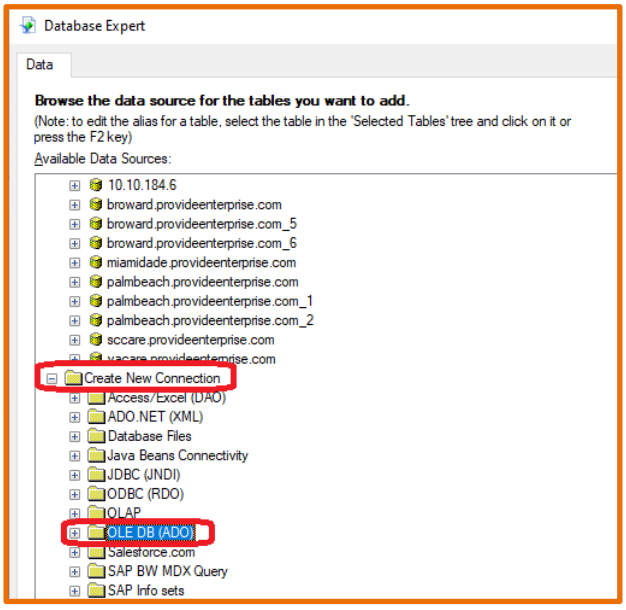
- When Crystal Reports starts up the welcome dialog opens by default. Either click the New Report button (circled at the top of the screenshot below) or select "Blank Report" under "START A NEW REPORT"as seen in the screenshot below.



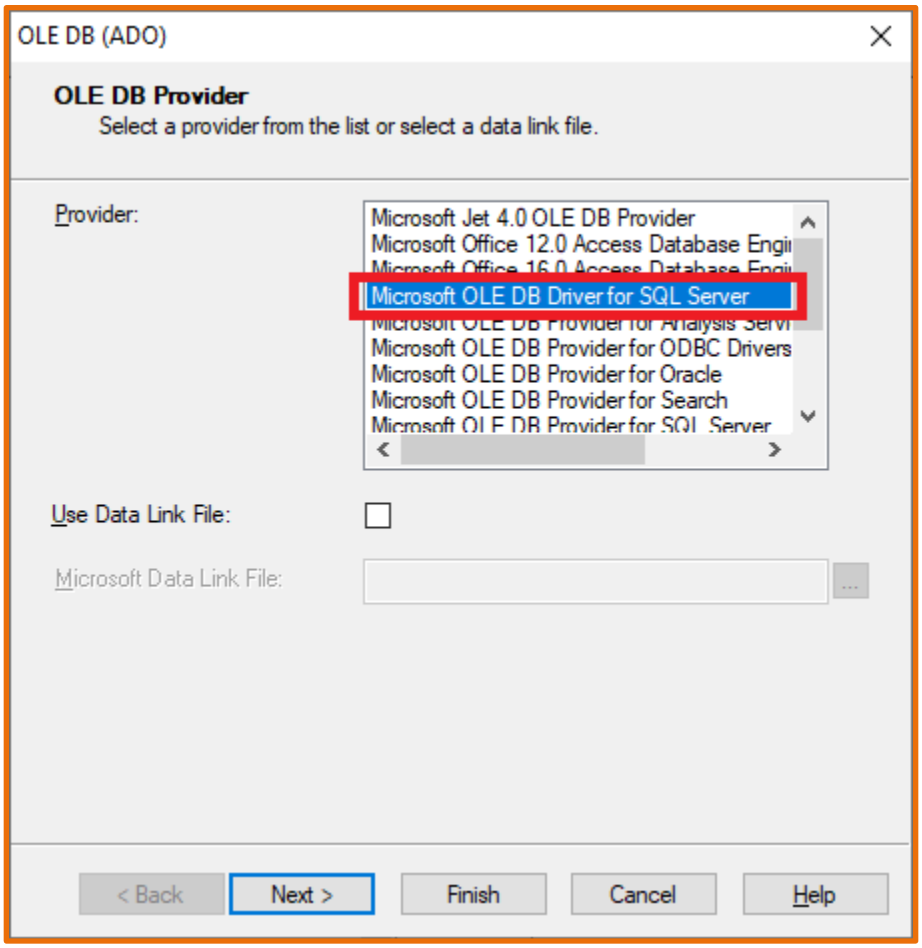
- This will activate the Database Expert as seen in the figure below:



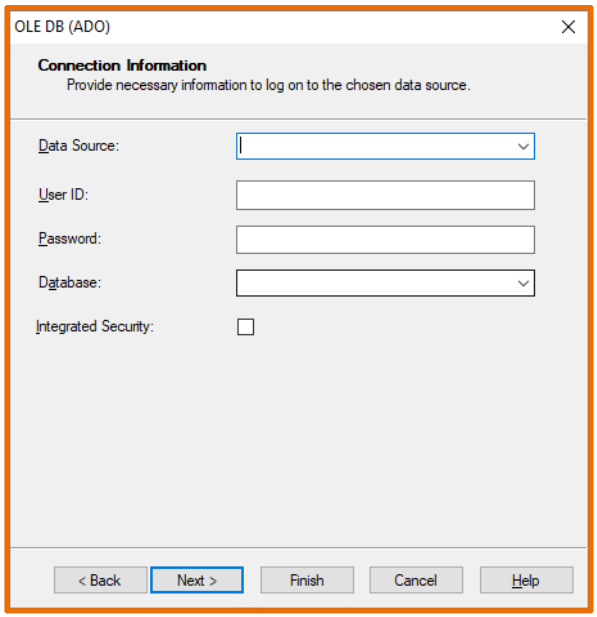
- Click on **Create New Connection** and then click the + before "OLE DB ADO".



- The following dialog box will then be activated:



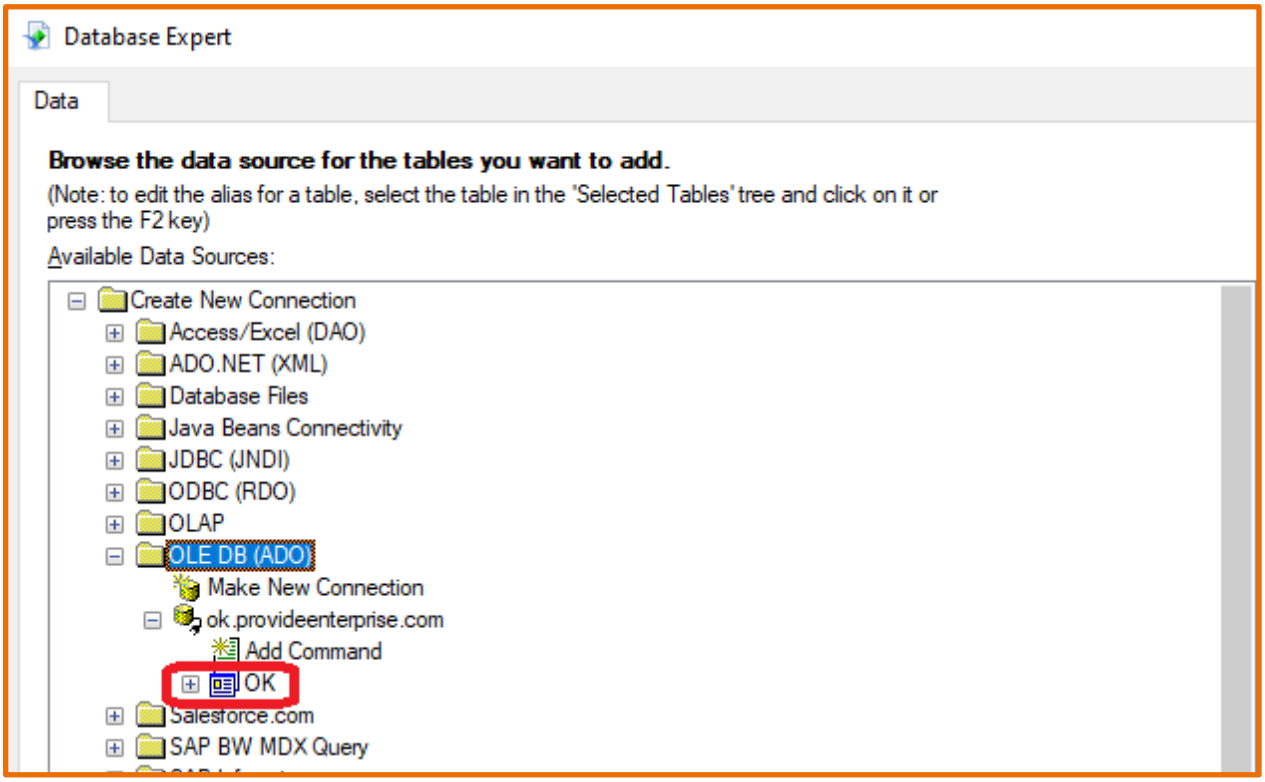
- Select **Microsoft OLE DB Driver for SQL Server** (shown in the above image).
- Click the **Next** button to continue. The following dialog will open:



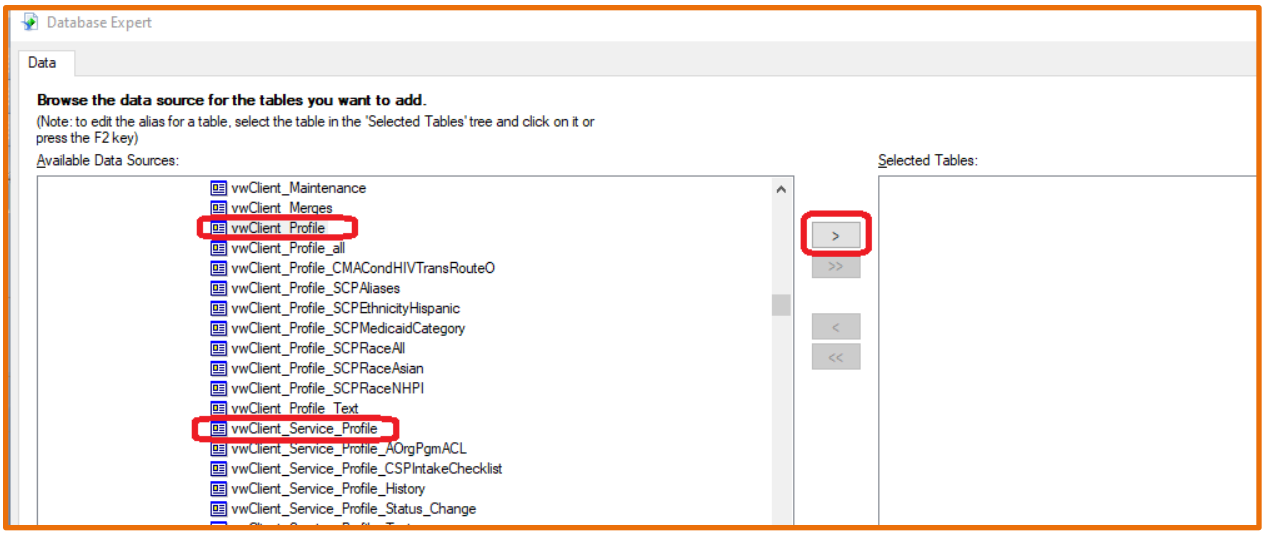
- Fill out the fields as specified below:

- **Data Source:** Enter the same **Server** that you use on your Provide Enterprise Login Screen.
- **User ID:** Enter your Provide Enterprise User Name.
- **Password:** Enter your current Provide Enterprise Password.
- **Database:** Enter the **Database** that you use on your Provide Enterprise Login Screen.

- Do **NOT** select "Integrated Security".
- Click the **Finish** Button to continue.
- If you entered the Server, Database, Username and Password correctly, your screen will refresh to show your new active connection to the server, similar to that in the figure below:



- Click the + in front of your database (circled in the figure above). This will expand the database to show the various tables and views available. Next, click the + in front of dbo and then click the + in front of Views. This will then display all of the views available for your reports, similar to that in the figure below:



- Scroll down the list until you see the views that you wish to add to the report (similar to those in the above image. Click on the view(s) you want to add to the report and then click on the right arrow in the middle (also shown in the above image).
- Once you have added all the necessary views for your report you should decide if this report is meant to capture all records of a certain type or a limited set of records. If it is meant to capture all records you can proceed with the design of your Crystal Report. If you want to limit the set of records you will want to add a SQL ID parameter field.

Using Parameter Fields and Selection Criteria in Embedded Crystal Report Objects

When embedded in Provide® Enterprise, Crystal Reports can be "called" from three different places. Client reports can be "called" from the Client Profile (Create Activity – Report) or from the Clients Views (if you wish to create the report on multiple Clients). Document reports can be "called" from individual forms in Provide® Enterprise, generally when attached to a "Report" button in the form. Typically, these are detailed "print" formats of the form. Stand Alone reports are not dependent on any particular document and can be run anywhere in the Provide® Enterprise Client, using the "Report – Run Report" function.

In the case of Client and Document reports, you must create a parameter field and a selection criteria in your Crystal Report so that the report "knows" which Client or document it should be associated with. In the section below, you will learn how to create this parameter field and the associated selection criteria.

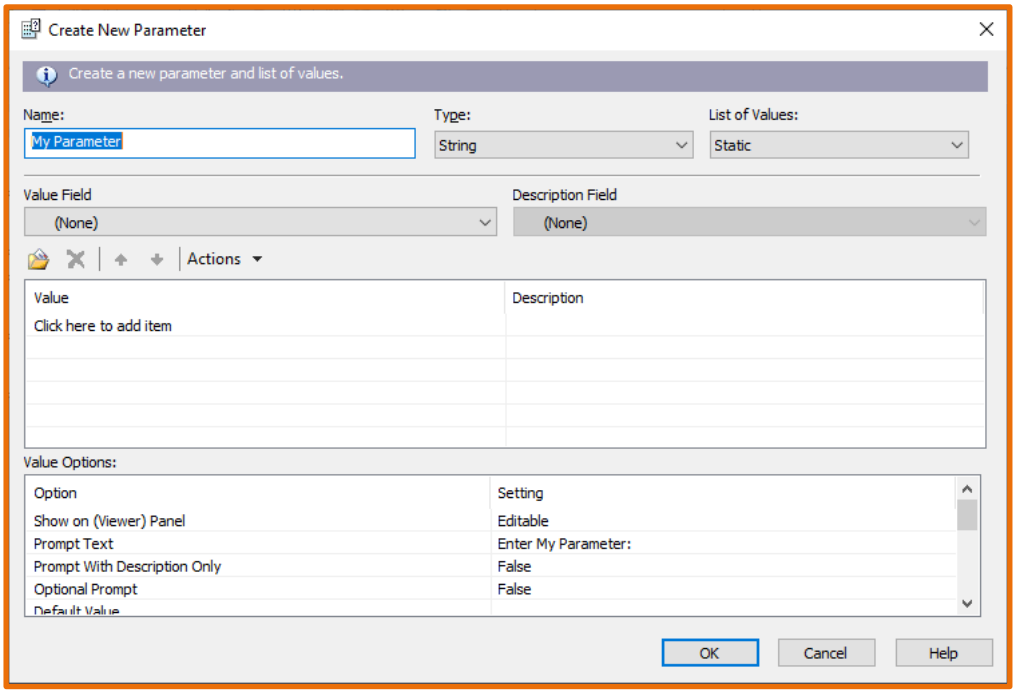
Client Parameter Fields and Selection Criteria

If your report is going to be "called" from the Client Profile or from the Clients views, you must add a parameter field and selection criteria item to your report utilizing the ClientProfileID (the Provide® Enterprise assigned unique identifier for each Client).

Creating a Parameter Field

To add a parameter field, follow these steps:

- Within your report, locate the Field Explorer. Under "Parameter Fields" right click and select "New".
- The new Parameter field will open similar to that in the figure below:



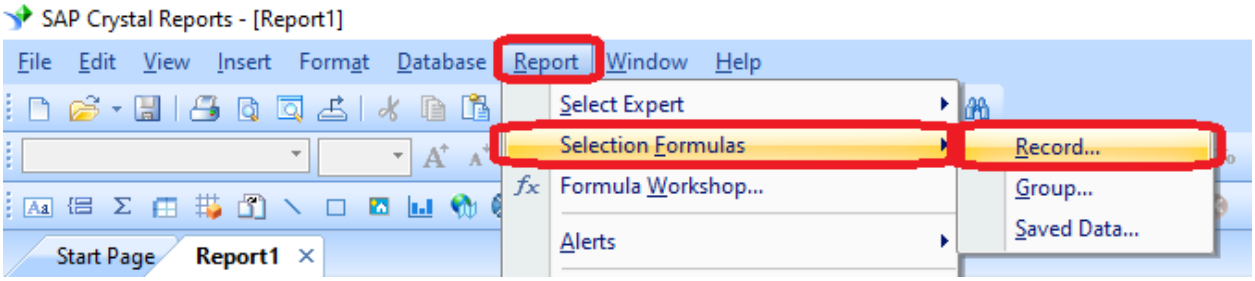
- Fill out the "Name" and "Type" fields as discussed below. If the parameter field is going to include multiple values (i.e. you are running the report on more than one Client at a time, make sure that Allow Multiple Values is selected under "Options".
- Click on the OK button to continue. Your parameter field will be saved.

Adding a Selection Criteria to the Report

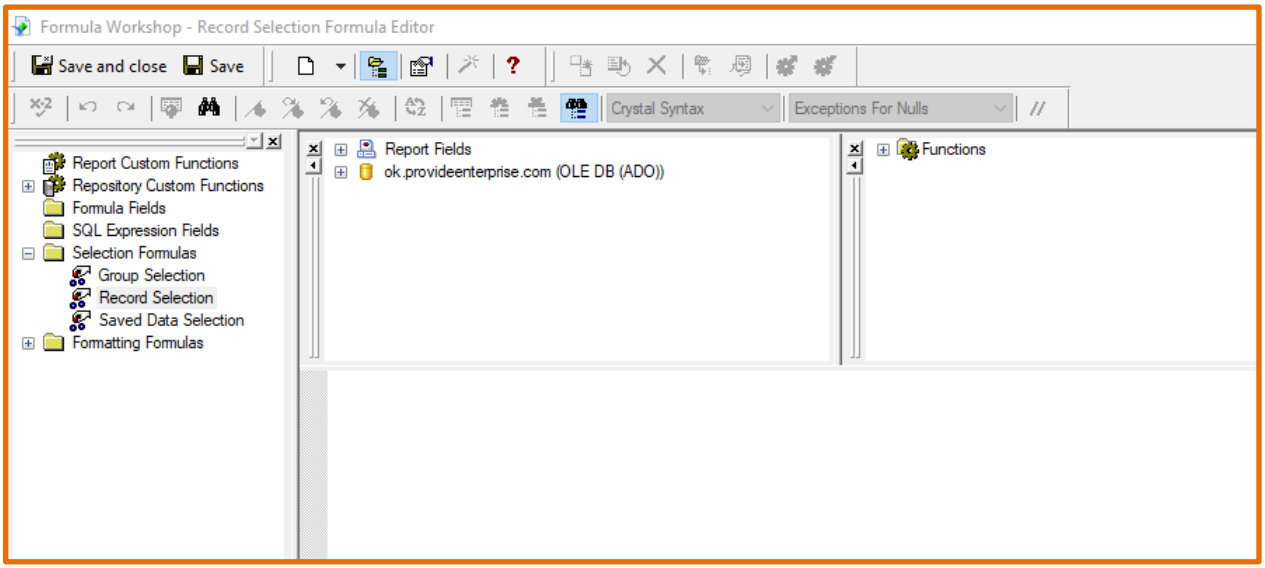
Once you have added your Parameter field, you are ready to include the parameter field in the Selection Criteria for your report.

To add the selection criteria, follow the steps below:

- From the Windows Toolbar, select "Report – Selection Formula - Record", circled in the following figure.



- This will bring up the Record Selection Formula Editor, seen in the figure below:.



- In the Criteria window at the bottom of the Formula Editor, enter your formula, as suggested below.
- Once done, click on the button to check your formula. If there are no errors, you can then click on the **Save and Close** button to save and close the Formula Editor.

Suggested Parameter Field Prompts and Selection Criteria – Client Reports

If you are going to use the Report as a Client report (discussed above), you should fill out the Parameter Field Prompts and the Selection Criteria as listed below:

Parameter Field Prompts

- Name: CPIDs
- Value Type: String
- Allow Multiple Values: Select this if you want to be able to run this report on multiple Clients at a time.

Selection Criteria Formula

{?CPIDs} = {vwClient_Profile.ClientProfileID}

Parameter Field Name = ClientProfileID from the form the report is being called from.

Suggested Parameter Field Prompts and Selection Criteria – Document Reports

If you are going to use the Report as a Document report (discussed above), you should fill out the Parameter Field Prompts and the Selection Criteria as listed below:

Parameter Field Prompts

Name: SQLID

Value Type: String

Selection Criteria Formula

{?SQLID} = totext({vwLetter.SQLID})

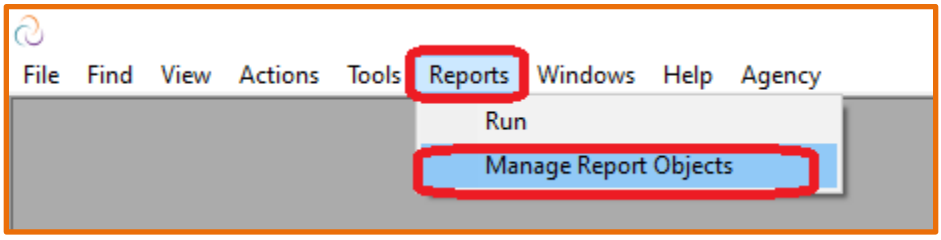
Parameter Field Name = totext (SQLID from the form the report is being called from.)

Chapter 5: Managing Crystal Report Objects in Provide® Enterprise

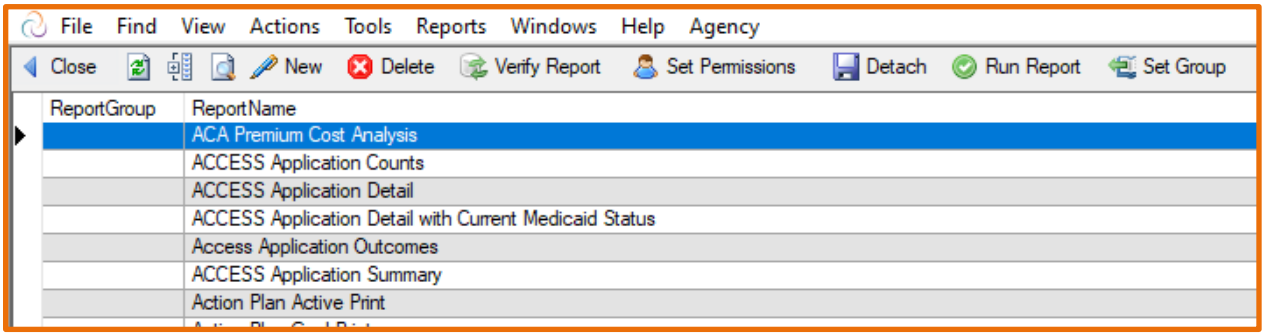
Managing Report Objects is generally a task reserved for System Administrators.

To do this, follow the steps below:

- Select the "Reports – Manage Report Objects" menu as shown below.



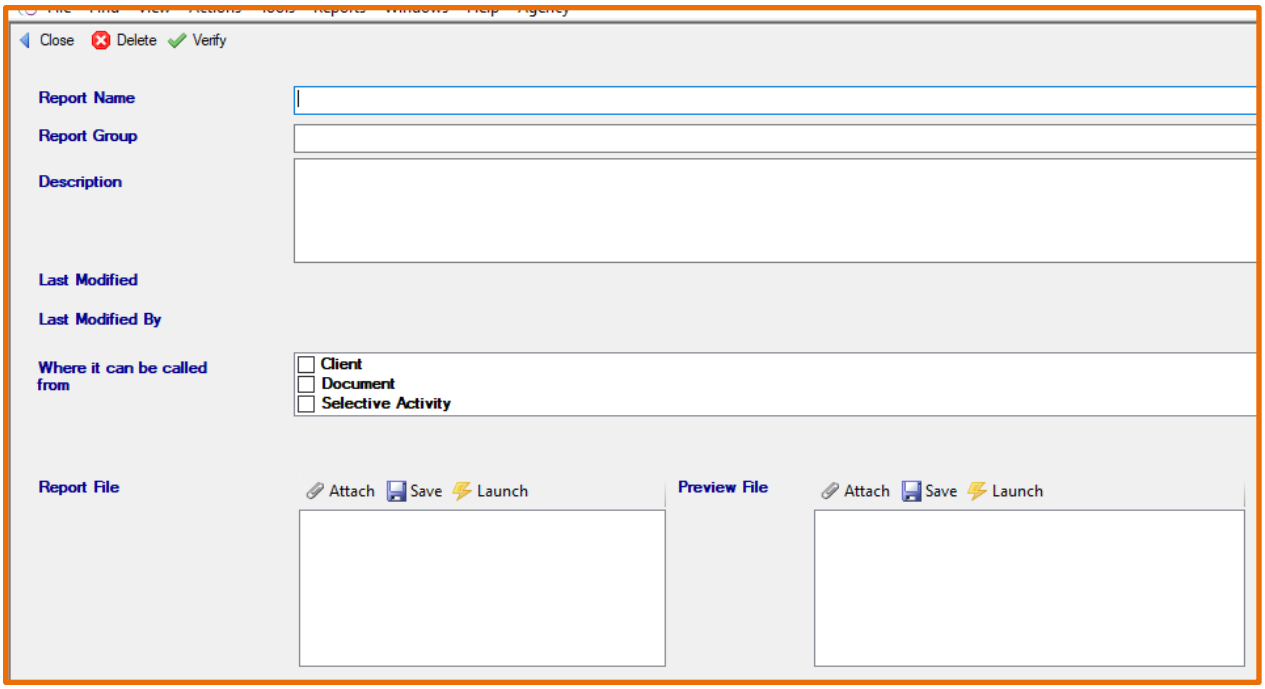
- A screen similar to the following will appear:



Adding a New Report Object

To add a new report object, follow the steps below:

- Click on the New button. You will then see a screen similar to the one below:



- Type in the report name and description in the applicable fields.
- Select the places the report can be called from by checking the desired boxes.
    - **Client:** The report will be available from the Create Activity\Report or Create Report button from in the Client Profile. The button will automatically push the ClientProfileID parameter to the report template.
    - **Document:** The report can be called from any document. Typically this is used if you wish to create a printable version of the form that looks different than it does on screen. The button will automatically push the SQL ID parameter into the report template.
    - **Stand Alone:** The report will be available from the Reports - Run selection to the Provide User Roles selected under the **Report Security** section of this form.
- Attach the report file by browsing and selecting it with the Attach button under the "Report File" section.
- You can also attach a preview copy of the report (for example, in a pdf format) that users can view prior to running the report. This is done also by selecting the attach button in the Preview File section.
- If the Report is being called from the "Client" or is a "Standalone" report, you will see a **Report Security** section to the report form, seen in the figure below:

- Select which User Role(s) should have the ability to run the new report.
- Click the **Close** button to save the report and make it available for use. The PE system will verify that all fields and tables in the report are available in the database and are appropriately named prior to the report saving. If the report meets the verification criteria, the report form will close and you will be returned to the Report list.

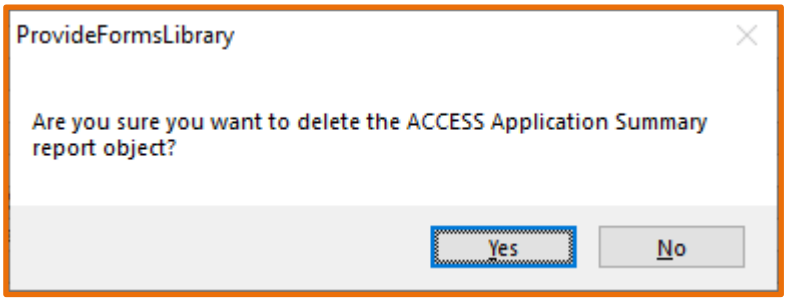
Deleting a Report Object

If you wish to delete a report object from the database, follow the steps below:
From the view of Report Objects (Reports – Manage Report Objects) single click on the report you wish to delete. (Or, you can double click on the report you wish to delete).

- Click on the

button.

A message similar to that below will appear:



- If you wish to delete the report, click on the **YES** button. If not, click **No.**
- If you select Yes, the report will be permanently deleted from the system.