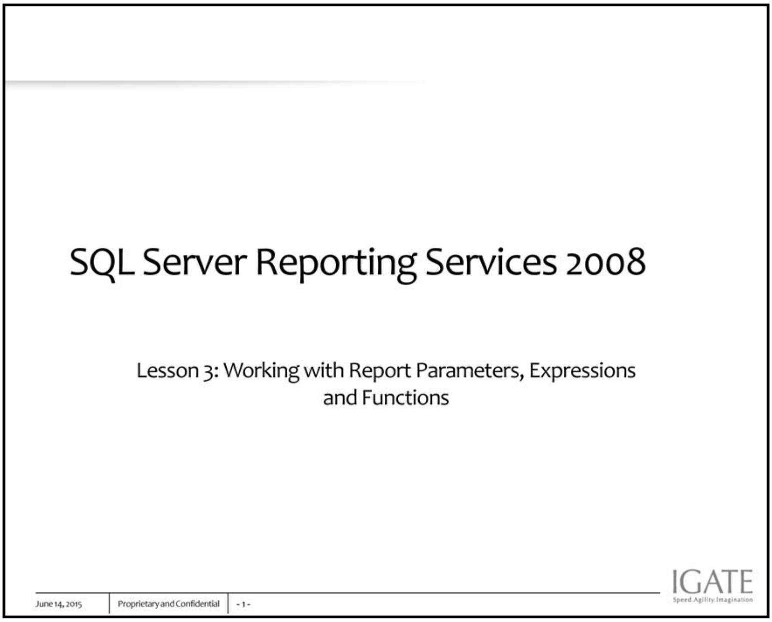
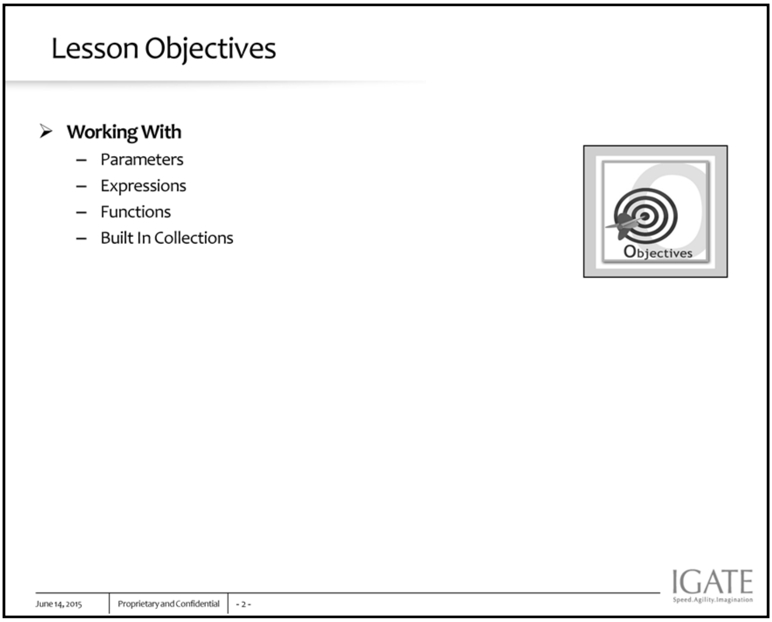


SQL Server Reporting Services 2008

Lesson 3: Working with Report Parameters, Expressions and Functions

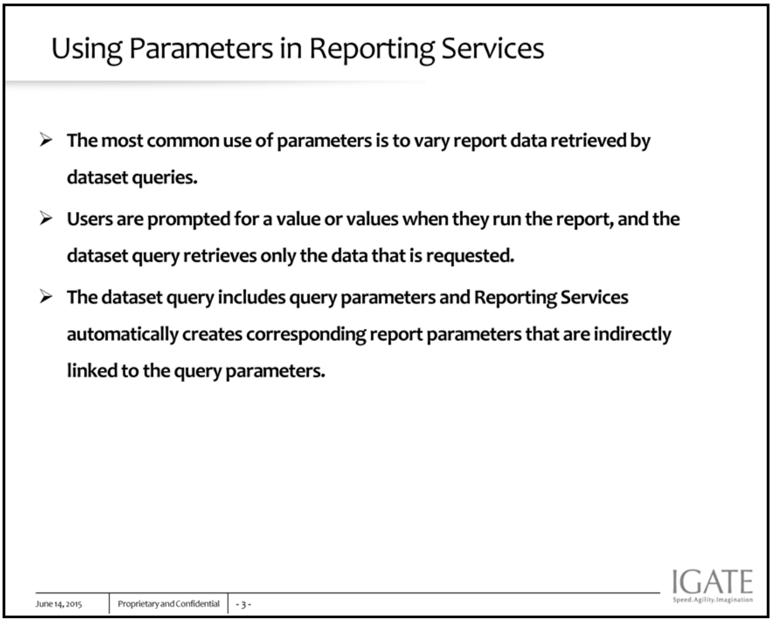
June 14, 2015 | Proprietary and Confidential | - 1 -



Lesson Objectives

- **Working With**
  - Parameters
  - Expressions
  - Functions
  - Built In Collections

June 14, 2015 | Proprietary and Confidential | - 2 -



Using Parameters in Reporting Services

- The most common use of parameters is to vary report data retrieved by dataset queries.
- Users are prompted for a value or values when they run the report, and the dataset query retrieves only the data that is requested.
- The dataset query includes query parameters and Reporting Services automatically creates corresponding report parameters that are indirectly linked to the query parameters.

June 14, 2015 | Proprietary and Confidential | - 3 -

IGATE Speed.Agility.Imagination

In Reporting Services, parameters are used to specify the data to use in a report. Report parameters typically filter report data that is retrieved from a data source. Filtering data at the data source can improve performance for processing and viewing a report.

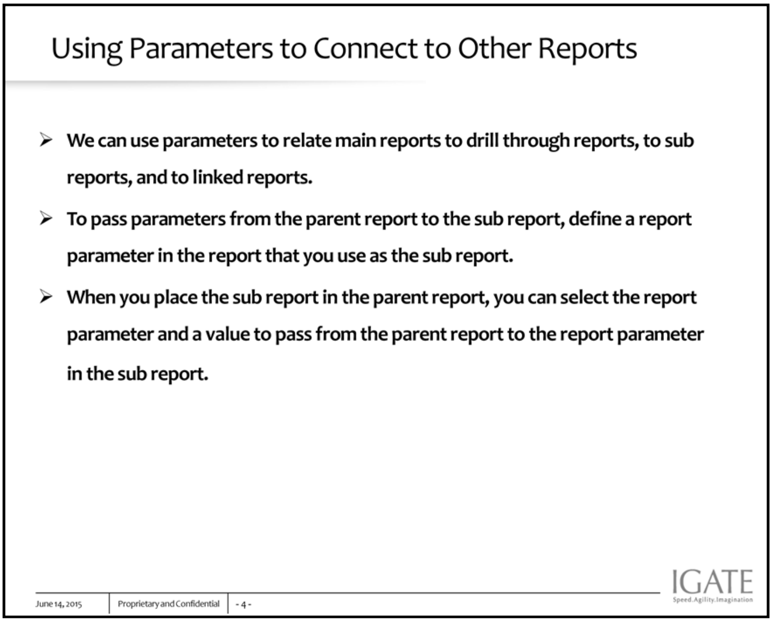
After a report parameter is created, you can modify the default values and other parameter properties, such as visibility. For example, report parameters can use built-in variables like UserID, so you might want to create a report with a hidden parameter that selects data specific to the user running the report. Parameters can contain single values or multiple values, use a static or query-based valid values list, and accept null or blank values. Multivalue parameters enable users to select more than one value at run time.

Dependent or Cascading Parameters
When you create a query that uses multiple query parameters, you can create a set of cascading parameters. Cascading parameters provide a way of filtering a very large number of parameter values down to a manageable number of values. For example, suppose a query includes the parameters @Category, @Subcategory, and @Product, where the list of subcategories is dependent on @Category, and the list of products is dependent on the @Subcategory. When a user chooses a value for the report parameter Category, the values for Subcategory are limited to valid values for the chosen category. After the user selects a value for Subcategory, the choices for Product have already been filtered by the choice for category and subcategory. Using this technique, you can reduce the valid choices for a parameter down to a reasonable number of values.

To design cascading parameters, you must include the following items in your report:

A. The main dataset query, which has multiple related query parameters.

B. separate dataset for each report parameter that supplies its available values. It is important to use the same case-sensitive spelling for each query parameter so that the query parameters and report parameters are linked properly. The query for each set of available values for each report parameter must provide only values that make sense in the context of the main query.



Using Parameters to Connect to Other Reports

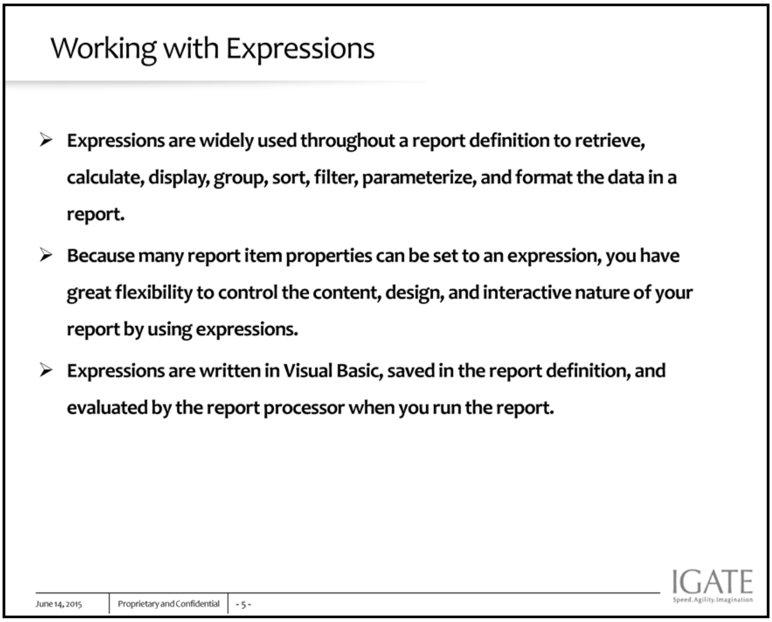
- **We can use parameters to relate main reports to drill through reports, to sub reports, and to linked reports.**
- **To pass parameters from the parent report to the sub report, define a report parameter in the report that you use as the sub report.**
- **When you place the sub report in the parent report, you can select the report parameter and a value to pass from the parent report to the report parameter in the sub report.**

June 14, 2015 | Proprietary and Confidential | - 4 -

Speed.Agility.Imagination

For dividing the report requirement, the sub reports could be used. A existing report also can be re used across other report designs.

Sub report could be linked with a parameter or be unlinked.

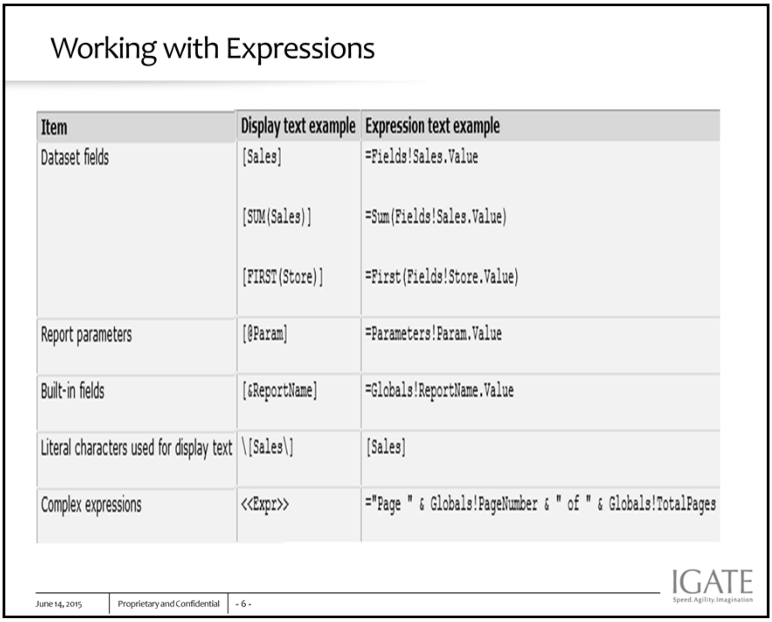


Working with Expressions

- **Expressions are widely used throughout a report definition to retrieve, calculate, display, group, sort, filter, parameterize, and format the data in a report.**
- **Because many report item properties can be set to an expression, you have great flexibility to control the content, design, and interactive nature of your report by using expressions.**
- **Expressions are written in Visual Basic, saved in the report definition, and evaluated by the report processor when you run the report.**

June 14, 2015 | Proprietary and Confidential | - 5 -

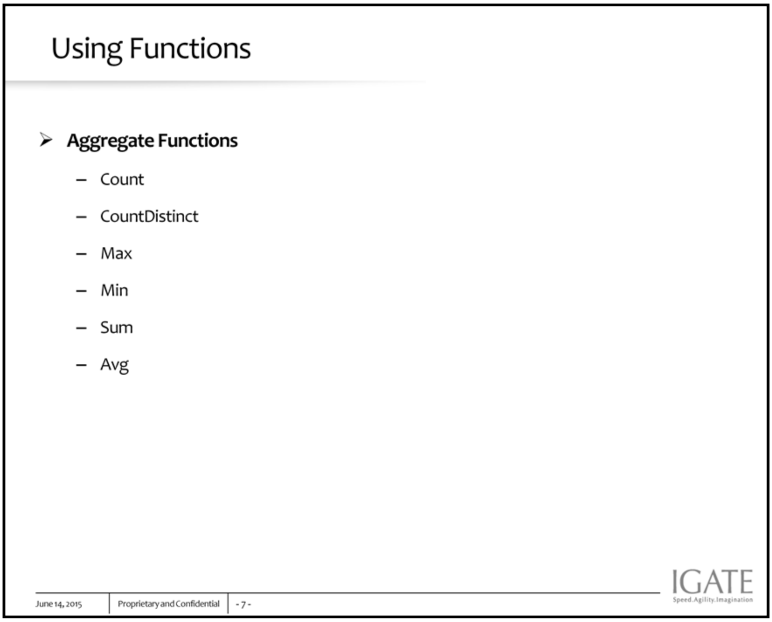
On the report design surface, expressions appear as simple or *complex expressions*. *Simple expressions* contain a reference to a single dataset field, parameter, or built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets; for example, a dataset field appears as [ProductID]. Simple expressions are created for you automatically (for example, when you drag a field from a dataset onto a text box), or you can type them directly into a data region cell, or text box on the design surface or in a dialog box. Complex expressions can contain multiple built-in references, operators, and function calls, and appear on the design surface as <<Expr>>. To see or change the expression text, you must open the **Expression** dialog box.



Working with Expressions

Item	Display text example	Expression text example
Dataset fields	[Sales] [SUM(Sales)] [FIRST(Store)]	=Fields!Sales.Value =Sum(Fields!Sales.Value) =First(Fields!Store.Value)
Report parameters	[@Param]	=Parameters!Param.Value
Built-in fields	[&ReportName]	=Globals!ReportName.Value
Literal characters used for display text	\[Sales\]	[Sales]
Complex expressions	<<Expr>>	="Page " & Globals!PageNumber & " of " & Globals!TotalPages

June 14, 2015 | Proprietary and Confidential | - 6 -



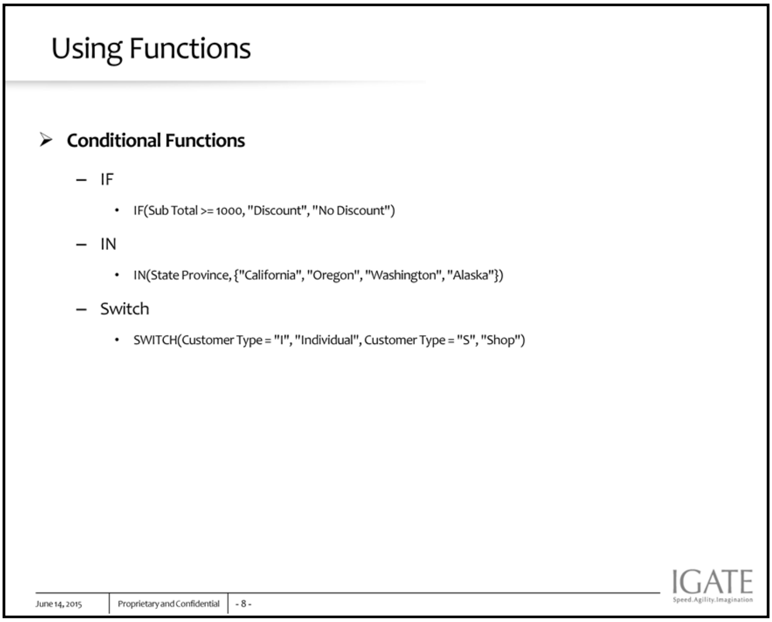
Using Functions

- **Aggregate Functions**
  - Count
  - CountDistinct
  - Max
  - Min
  - Sum
  - Avg

June 14, 2015 | Proprietary and Confidential | - 7 -

Functions are predefined formulas that perform calculations by using specific values, called arguments, in a particular order. Functions can be used to perform simple or complex calculations. The structure of a function begins with the function name, followed by an opening parenthesis, the arguments for the function separated by commas, and a closing parenthesis.

Arguments can be field references, numbers, text, and logical values such as TRUE or FALSE. Arguments can also be constants, formulas, or other functions. The arguments that you enter must produce a valid value for that argument. For example, if the formula is multiplying two integers, the result cannot be a text string.



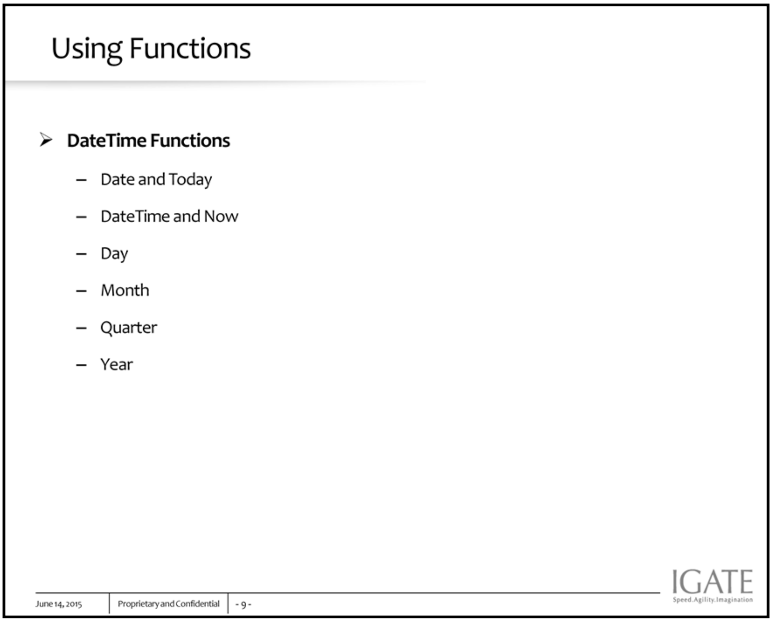
Using Functions

- **Conditional Functions**
  - IF
    - IF(Sub Total >= 1000, "Discount", "No Discount")
  - IN
    - IN(State Province, {"California", "Oregon", "Washington", "Alaska"})
  - Switch
    - SWITCH(Customer Type = "I", "Individual", Customer Type = "S", "Shop")

June 14, 2015 | Proprietary and Confidential | - 8 -

IGATE
Speed.Agility.Imagination

On the report design surface, expressions appear as simple or *complex expressions*. *Simple expressions* contain a reference to a single dataset field, parameter, or built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets; for example, a dataset field appears as [ProductID]. Simple expressions are created for you automatically (for example, when you drag a field from a dataset onto a text box), or you can type them directly into a data region cell, or text box on the design surface or in a dialog box. Complex expressions can contain multiple built-in references, operators, and function calls, and appear on the design surface as <<Expr>>. To see or change the expression text, you must open the **Expression** dialog box.



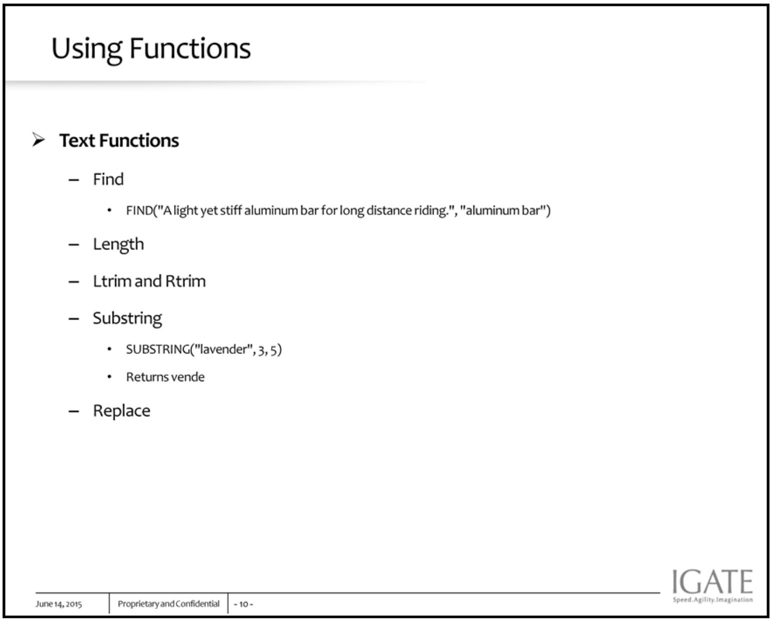
Using Functions

- **DateTime Functions**
  - Date and Today
  - DateTime and Now
  - Day
  - Month
  - Quarter
  - Year

June 14, 2015 | Proprietary and Confidential | - 9 -

IGATE Speed.Agility.Imagination

On the report design surface, expressions appear as simple or *complex expressions*. *Simple expressions* contain a reference to a single dataset field, parameter, or built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets; for example, a dataset field appears as [ProductID]. Simple expressions are created for you automatically (for example, when you drag a field from a dataset onto a text box), or you can type them directly into a data region cell, or text box on the design surface or in a dialog box. Complex expressions can contain multiple built-in references, operators, and function calls, and appear on the design surface as <<Expr>>. To see or change the expression text, you must open the **Expression** dialog box.



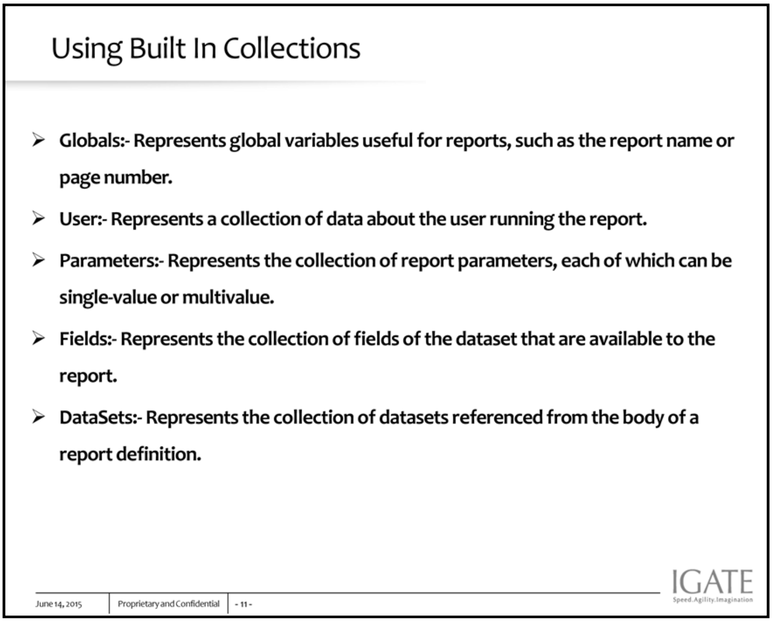
Using Functions

- **Text Functions**
  - Find
    - FIND("A light yet stiff aluminum bar for long distance riding.", "aluminum bar")
  - Length
  - Ltrim and Rtrim
  - Substring
    - SUBSTRING("lavender", 3, 5)
    - Returns vende
  - Replace

June 14, 2015 | Proprietary and Confidential | - 10 -

IGATE
Speed.Agility.Imagination

On the report design surface, expressions appear as simple or *complex expressions*. *Simple expressions* contain a reference to a single dataset field, parameter, or built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets; for example, a dataset field appears as [ProductID]. Simple expressions are created for you automatically (for example, when you drag a field from a dataset onto a text box), or you can type them directly into a data region cell, or text box on the design surface or in a dialog box. Complex expressions can contain multiple built-in references, operators, and function calls, and appear on the design surface as <<Expr>>. To see or change the expression text, you must open the **Expression** dialog box.



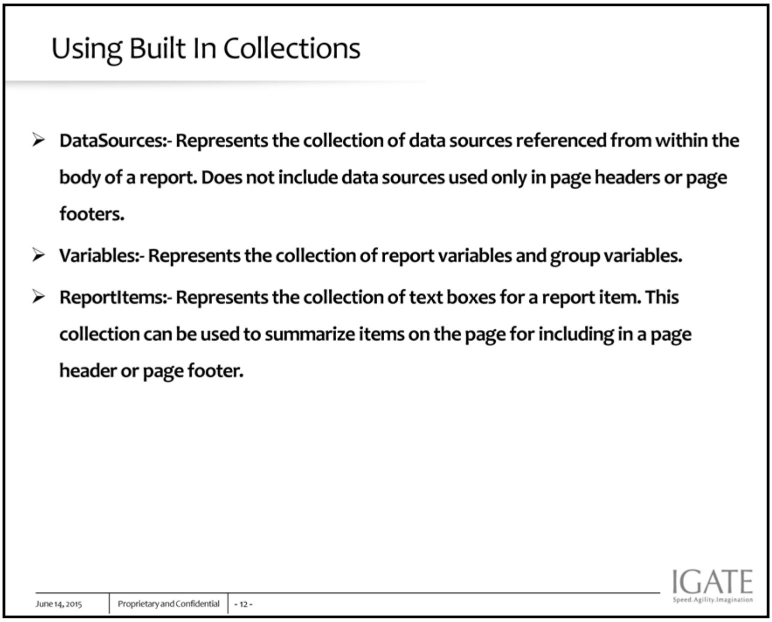
Using Built In Collections

- Globals:- Represents global variables useful for reports, such as the report name or page number.
- User:- Represents a collection of data about the user running the report.
- Parameters:- Represents the collection of report parameters, each of which can be single-value or multivalue.
- Fields:- Represents the collection of fields of the dataset that are available to the report.
- DataSets:- Represents the collection of datasets referenced from the body of a report definition.

June 14, 2015 | Proprietary and Confidential | - 11 -

IGATE Speed.Agility.Imagination

Reporting Services provides the following built-in collections that you can reference from an expression: ReportItems, Parameters, Fields, DataSets, DataSources, Variables, and built-in fields for global information such as the report name. Not all collections appear in the **Expression** dialog box. The DataSets and DataSources collections are available only at run-time for published reports on a report server. The ReportItems collection is collection of text boxes in a report region, for example, the text boxes on a page or in a page header.

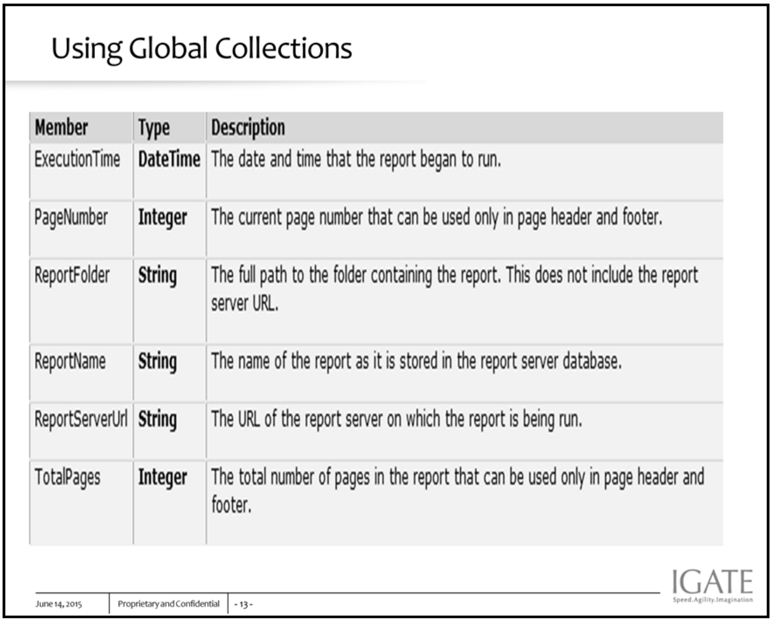


Using Built In Collections

- **DataSources:- Represents the collection of data sources referenced from within the body of a report. Does not include data sources used only in page headers or page footers.**
- **Variables:- Represents the collection of report variables and group variables.**
- **ReportItems:- Represents the collection of text boxes for a report item. This collection can be used to summarize items on the page for including in a page header or page footer.**

June 14, 2015 | Proprietary and Confidential | - 12 -

IGATE Speed.Agility.Imagination



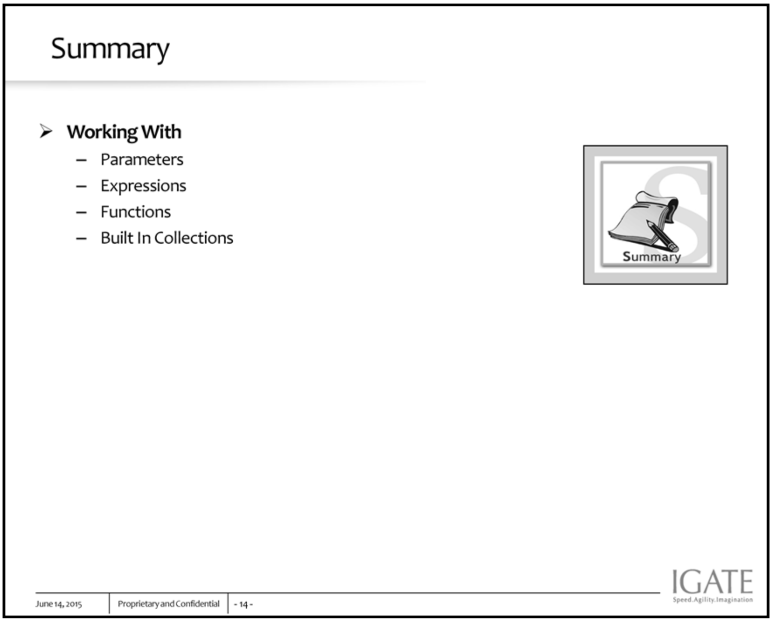
Using Global Collections

Member	Type	Description
ExecutionTime	**DateTime**	The date and time that the report began to run.
PageNumber	**Integer**	The current page number that can be used only in page header and footer.
ReportFolder	**String**	The full path to the folder containing the report. This does not include the report server URL.
ReportName	**String**	The name of the report as it is stored in the report server database.
ReportServerUrl	**String**	The URL of the report server on which the report is being run.
TotalPages	**Integer**	The total number of pages in the report that can be used only in page header and footer.

June 14, 2015 | Proprietary and Confidential | - 13 -

IGATE
Speed.Agility.Imagination

The Built-in fields collection, which includes both the **Globals** and the **User** collections, represent global values provided by Reporting Services when a report is processed. The **Globals** collection provides values such as the name of the report, the time when report processing began, and current page numbers for the report header or footer.



Summary

- **Working With**
  - Parameters
  - Expressions
  - Functions
  - Built In Collections

June 14, 2015 | Proprietary and Confidential | - 14 -