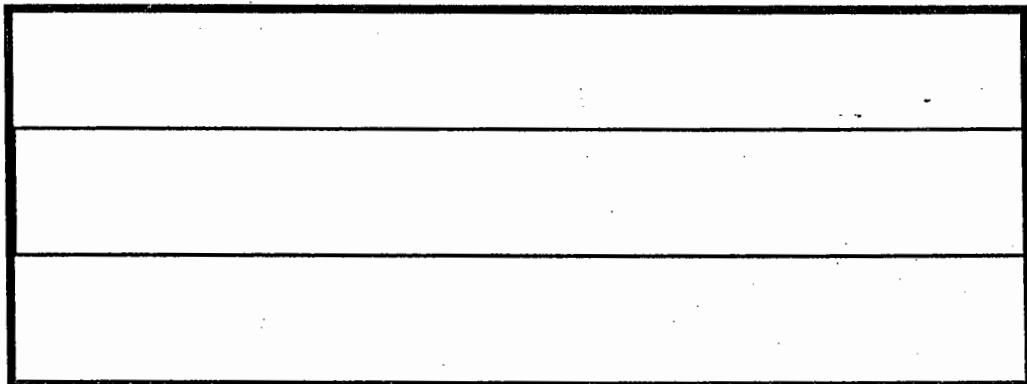


RS = 105/-

# SSIS & SSRS



Sri  TechHouse

# 608-A, Nilgiri Block,  
Aditya Enclave, Ameerpet, Hyd.

Ph : 040-66638869

## Questions in interview SSRS :

1. Tablix properties
2. Cascade parameters
3. Grouping and group properties
- \* 4. Drill down, drill through.
5. Explain Gauge, indicators, databases, sparkline.
6. Diff. b/w Table and Matrix.

## from Report Manager

1. Explain about Report Builder.
2. Explain about Report processing options (Cache / snapshot).
3. Explain about Subscriptions
4. Diff. b/w Report Manager Report Server.

## SSIS FAQ's

1. How many engines SSIS use?
2. Why we call dataflow process as IN MEMORY process?
3. Diff. b/w Execute SQL task and OLEDB source?

SQL Server 2012 is Microsoft's latest cloud-ready information platform. Organizations can use SQL Server 2012 to efficiently protect, unlock, and scale the power of their data across the desktop, mobile device, datacenter, and either a private or public cloud. Building on the success of the SQL Server 2008 R2 release, SQL Server 2012 has made a strong impact on organizations worldwide with its significant capabilities. It provides organizations with mission-critical performance and availability, as well as the potential to unlock breakthrough insights with pervasive data discovery across the organization. Finally, SQL Server 2012 delivers a variety of hybrid solutions you can choose from. For example, an organization can develop and deploy applications and database solutions on traditional non virtualized environments, on appliances, and in on-premises private clouds or off-premises public clouds. Moreover, these solutions can easily integrate with one another, offering a fully integrated hybrid solution.

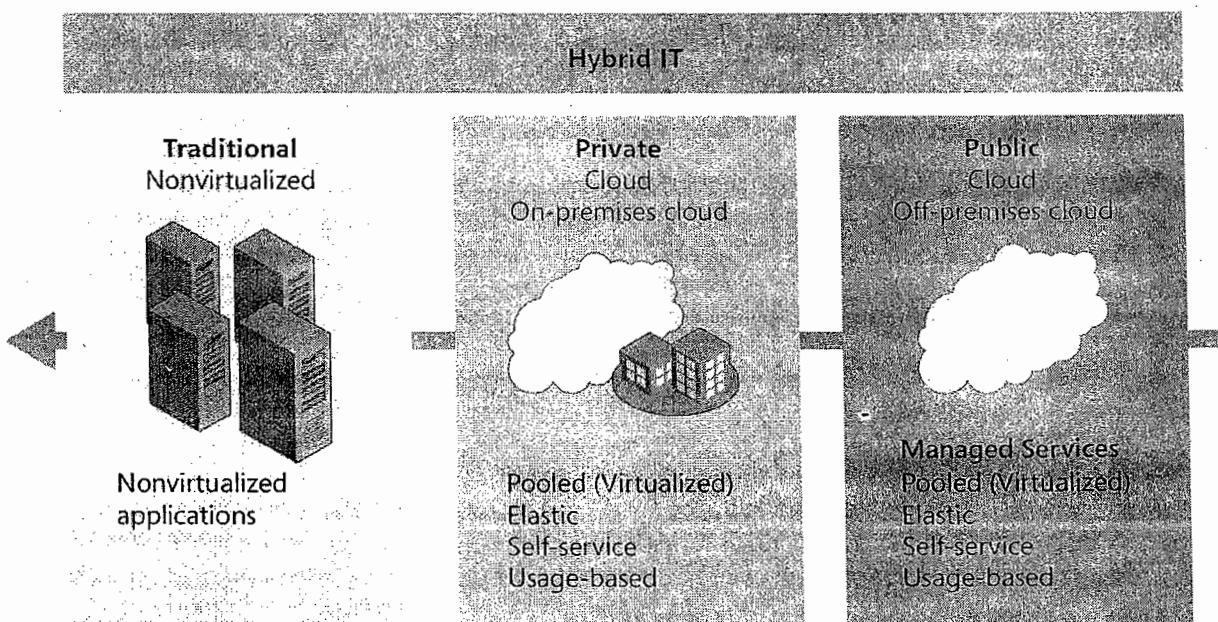


FIGURE 1-1 SQL Server 2012, cloud-ready information platform

**SQL Server 2012 Editions :** Three main editions.

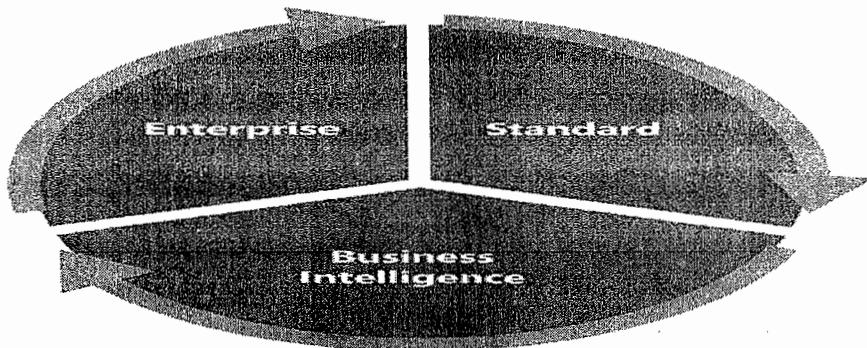


FIGURE 1-7 The main editions of SQL Server 2012

### BUSINESS INTELLIGENCE EDITION

For the first time in the history of SQL Server, a Business Intelligence edition is offered. The Business Intelligence edition offers organizations the full suite of powerful BI capabilities such as scalable reporting and analytics, Power View, and PowerPivot.

Here is a high-level list of what the new Business Intelligence edition includes:

- Up to a maximum of 16 cores for the Database Engine
- Maximum number of cores for business intelligence processing
- All of the features found in the Standard edition
- Corporate business intelligence
- Reporting
- Analytics
- Multidimensional BI semantic model
- Self-service capabilities
- Alerting
- Power View
- PowerPivot for SharePoint Server
- Enterprise data management
- Data quality services
- Master data services
- In-memory tabular BI semantic model
- Basic high availability can be achieved with AlwaysOn 2-Node Failover Clustering

### HARDWARE & SOFTWARE REQUIREMENTS

**TABLE 1-1** Hardware Requirements

Hardware Component	Requirements
Processor	Processor type: (64-bit) x64 Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support Processor speed: minimum 1.4 GHz; 2.0 GHz or faster recommended Processor type: (32-bit) Intel Pentium III-compatible processor or faster Processor speed: minimum 1.0 GHz; 2.0 GHz or faster recommended
Memory (RAM)	Minimum: 1 GB Recommended: 4 GB or more Maximum: Operating system maximum
Disk Space	Disk space requirements will vary depending on the components you install: Database Engine: 811 MB Analysis Services: 345 MB Reporting Services: 304 MB Integration Services: 591 MB Client components: 1823 MB

**TABLE 1-2** Software Requirements

Software Component	Requirements
Operating system	Windows Server 2008 R2 SP1 64-bit Datacenter, Enterprise, Standard or Web edition. or Windows Server 2008 SP2 64-bit Datacenter, Enterprise, Standard or Web edition
.NET Framework	Microsoft .NET Framework 3.5 SP1 and Microsoft .NET Framework 4.0

## SQL Server 2012 Prerequisites for Server Core

Organizations installing SQL Server 2012 on Windows Server 2008 R2 Server Core must meet the following operating system, features, and components prerequisites.

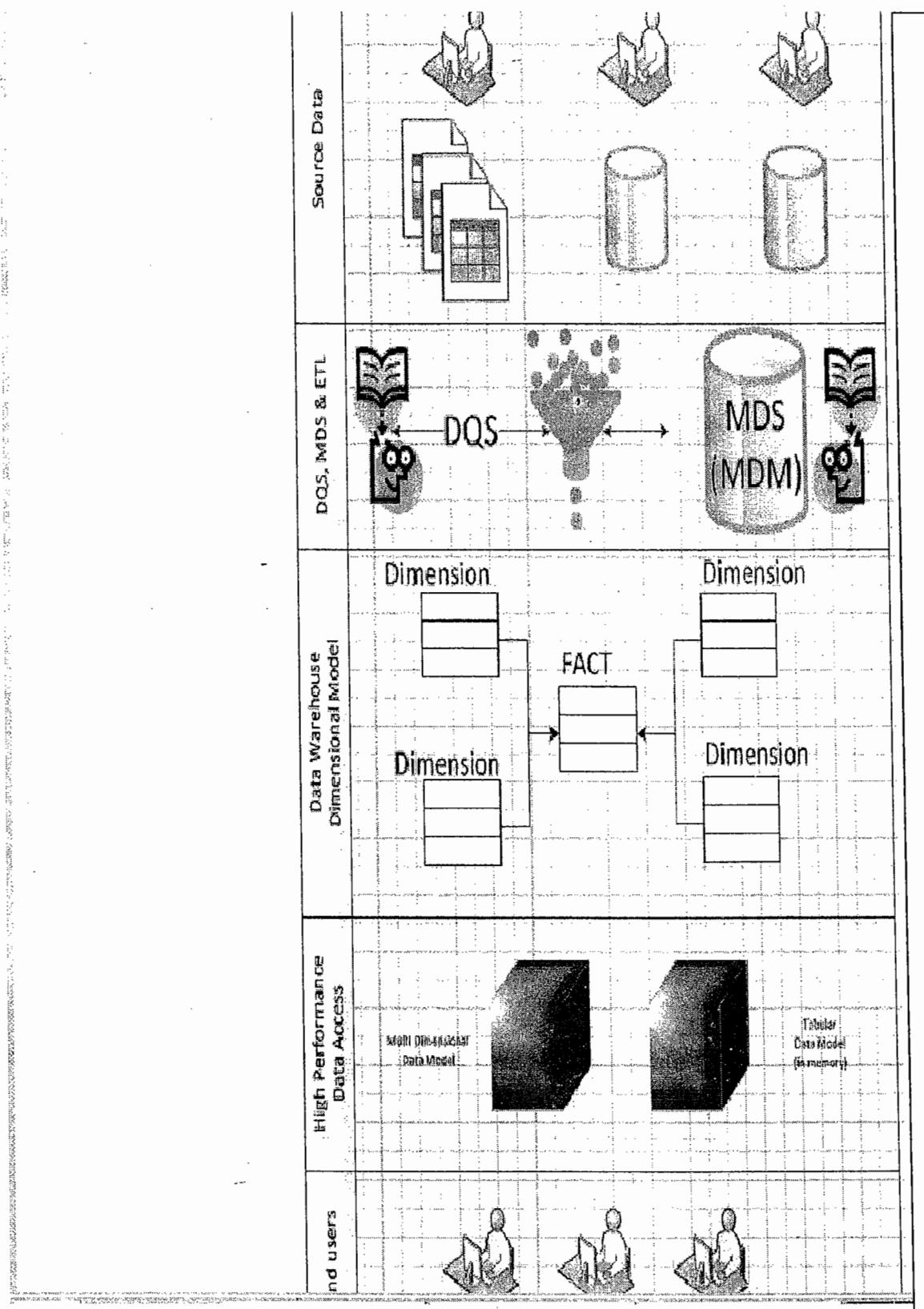
The operating system requirements are as follows:

- Windows Server 2008 R2 SP1 64-bit x64 Data Center Server Core
- Windows Server 2008 R2 SP1 64-bit x64 Enterprise Server Core
- Windows Server 2008 R2 SP1 64-bit x64 Standard Server Core
- Windows Server 2008 R2 SP1 64-bit x64 Web Server Core

Here is the list of features and components:

- .NET Framework 2.0 SP2
- .NET Framework 3.5 SP1 Full Profile
- .NET Framework 4 Server Core Profile
- Windows Installer 4.5
- Windows PowerShell 2.0

Once you have all the prerequisites, it's important to become familiar with the SQL Server components supported on Server Core.



## **Database Engine**

The Database Engine is the core service for storing, processing and securing data. The Database engine provides controlled access and rapid transaction processing to meet the requirements of the most demanding data consuming applications within your enterprise. The Database Engine also provides rich support for sustaining high availability.

## **Data Quality Services**

SQL Server Data Quality Services (DQS) provides you with a knowledge-driven data cleansing solution. DQS enables you to build a knowledge base, and then use that knowledge base to perform data correction and deduplication on your data, using both computer-assisted and interactive means. You can use cloud-based reference data services, and you can build a data management solution that integrates DQS with SQL Server Integration Services and Master Data Services.

## **Analysis Services**

Analysis Services is an analytical data platform and toolset for personal, team, and corporate business intelligence. Servers and client designers support traditional OLAP solutions, new tabular modeling solutions, as well as self-service analytics and collaboration using Excel, PowerPivot, and a SharePoint Server environment. Analysis Services also includes Data Mining so that you can uncover the patterns and relationships hidden inside large volumes of data.

## **Integration Services**

Integration Services is a platform for building high performance data integration solutions, including packages that provide extract, transform, and load (ETL) processing for data warehousing.

## **Master Data Services**

Master Data Services is the SQL Server solution for master data management. A solution built on Master Data Services helps ensure that reporting and analysis is based on the right information. Using Master Data Services, you create a central repository for your master data and maintain an auditable, securable record of that data as it changes over time.

**Replication** Replication is a set of technologies for copying and distributing data and database objects from one database to another, and then synchronizing between databases to maintain consistency. By using replication, you can distribute data to different locations and to remote or mobile users by means of local and wide area networks, dial-up connections, wireless connections, and the Internet.

**Reporting Services** Reporting Services delivers enterprise, Web-enabled reporting functionality so you can create reports that draw content from a variety of data sources, publish reports in various formats, and centrally manage security and subscriptions.

## DISCONTINUED INTEGRATION SERVICES FUNCTIONALITY IN 2012

The below features available in the older versions and removed in 2012.

- removed the direct interaction b/w control flow and dataflow.
- active-x scripting removed in 2012.
- Execute DTS task removed.
- Data viewing options such as histogram, scatter polar etc removed.
- Configuration functionality decreased, direct accessing of configurations removed from ssisr menu. etc.

### THE SQL SERVER DATA TOOLS EXPERIENCE

The **SQL Server Data Tools (SSDT)** was previously called **Business Intelligence Development Studio (BIDS)**, and it is the central environment in which you'll spend most of your time as an SSIS developer.

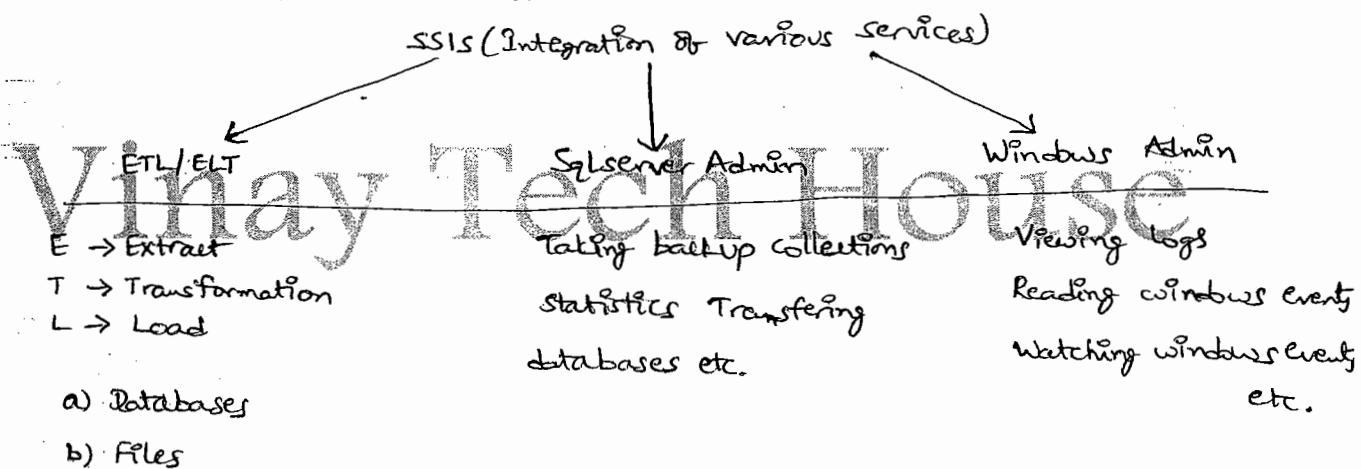
SSDT is just a specialized use of the familiar Visual Studio 2010 development environment. Visual Studio can host many different project types, from Console applications to Class Libraries and Windows applications. Although you may see many project types when you create a project, SSDT actually contains project templates for only Analysis Services, Integration Services, Report Server, and variants thereof.

SSIS in particular uses a BI project type called an Integration Services project, which provides a development design surface with a This development environment is similar to the legacy DTS Designer, but the approach is completely different. Most important, this is a collaborative development environment just like any Visual Studio development effort, with full source code management, version control, and multi-user project management.

**SSIS solutions are developed just like all other .NET development solutions**, including being persisted to files — in this case, XML file structures with a **.DSTX file** extension. You can even develop within the SSDT environment without a connection to a SQL Server instance using the offline mode. Once your solution is complete, it can be built and deployed to one or multiple target SQL servers. These changes from DTS to SSIS are crucial to establishing the discipline and best practices of existing software development methodologies as you develop business intelligence solutions.

## **SQLSERVER EVALUATION (SSIS HISTORY) & OPERATIONS**

- =====
- ① Till Sqlserver 7 ,there is no importing and exporting of data.  
DTS released in Sqlservr 7
  - ② Data Transformation Services  
(It supports the loading and unloading of OLEDB applications)  
↓
  - ③ **a** Sqlserver 2000 --They extended DTS with Active-X scripting (Most of the people impressed,DBAs habituated)
  - ③ **↓** After 5 years  
**SSIS** (All possible services)  
Easy usable,easy acceable,managable.  
( Full Support to GUI)



Ex: DB ↔ DB

DB ↔ File

File ↔ File

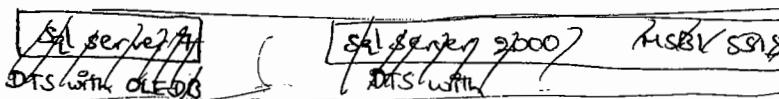
Databases :

Teradata, oracle ,

sql server etc .

Files

Flat, RAW, XML, Excel



**NOTE** Any application / program wants to communicate with the database always intermediary required (OLE/ODBC)

DTS -- Data Transformation Server

Designed for Load and unload operations

**SSIS OPPORTUNITIES**

**SSIS DEVELOPER RESPONSIBILITIES**

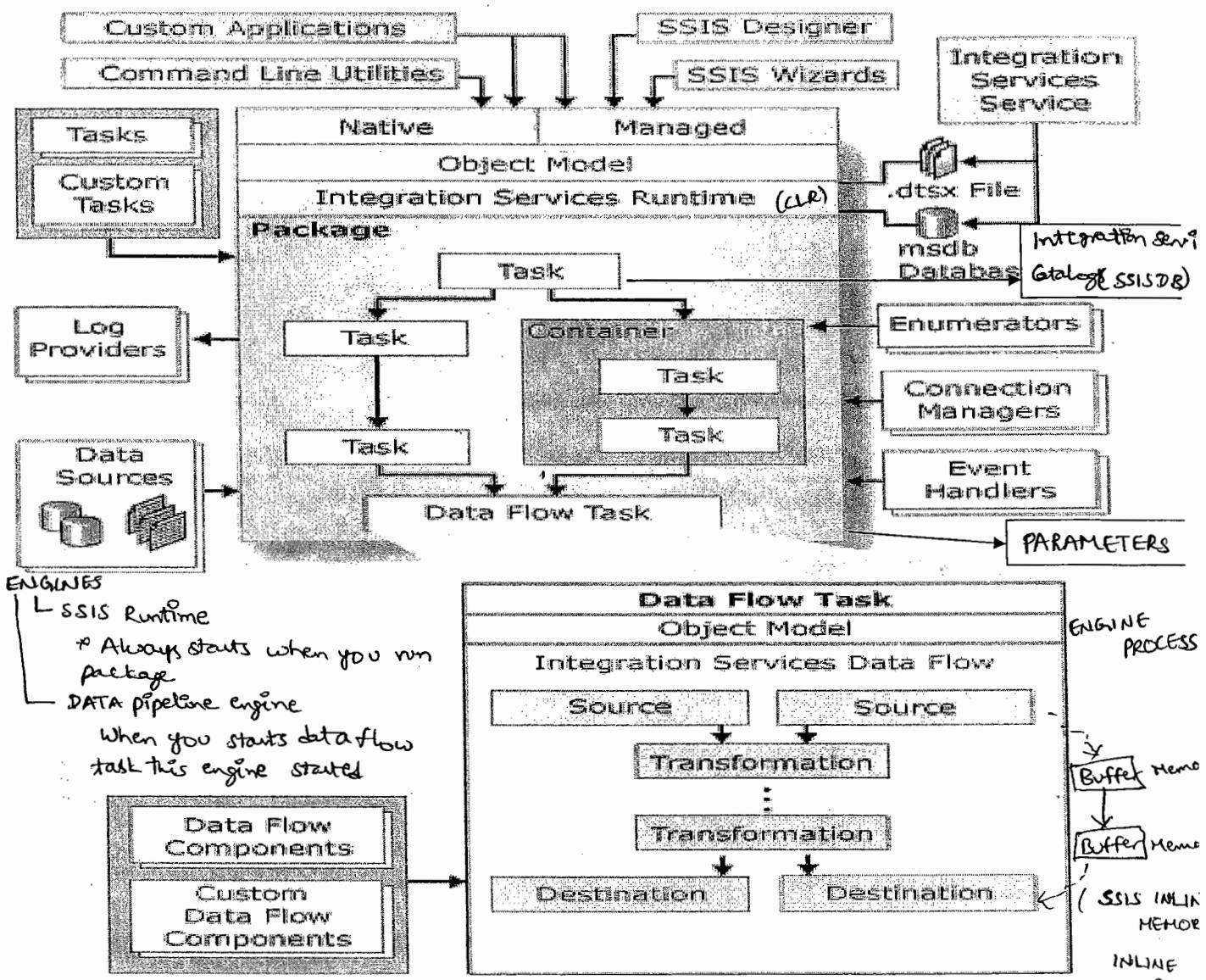
**Vinay Tech House**

**SSIS DBA RESPONSIBILITIES**

**SSIS SUPPORT /MAINTENANCE RESPONSIBILITIES**

**SSIS MODELLER /DESIGNER RESPONSIBILITIES**

## SSIS LOGICAL ARCHITECTURE



Of the components shown in the previous diagram, here are some important components.

### SSIS Designer

SSIS Designer is a graphical tool that you can use to create and maintain Integration Services packages. SSIS Designer is available in SSDT (SSDT earlier BIDS) (Business Intelligence Development Studio) as part of an Integration Services project.

### Runtime Engine

The Integration Services runtime saves the layout & packages, runs packages and provides support for logging, breakpoint, configurations, connection and transactions.

## **Integration Services Server in SQL Server Management Studio**

When you connect to an instance of the SQL Server Database Engine that hosts the SSISDB database, you see the following objects in Object Explorer:

- **SSISDB Database**

The SSISDB database appears under the Databases node in Object Explore. You can

*query the views and call the stored procedures that manage integration services service (server) and the objects that are stored on the server.*

- **Integration Services Catalogs**

Under the Integration Services Catalogs node there are folders for Integration Services projects and environments.

## **Tasks and Other Executables**

The Integration Services run-time executables are the package, containers, tasks, and event handlers that Integration Services includes. Run-time executables also include custom tasks that you develop.

## **Data Flow Engine (Also known as pipeline engine) and Data Flow Components**

The Data Flow task encapsulates the data flow engine. The data flow engine provides the in-memory buffers that move data from source to destination, and calls the sources that extract data from files and relational databases. The data flow engine also manages the transformations that modify data, and the destinations that load data or make data available to other processes. Integration Services data flow components are the sources, transformations, and destinations that Integration Services includes. You can also include custom components in a data flow.

## **API or Object Model**

The Integration Services object model includes managed application programming interfaces (API) for creating custom components for use in packages, or custom applications that create, load, run, and manage packages. Developer can write custom applications or custom tasks or transformations by using any common language runtime (CLR) compliant language.

## **Integration Services Service**

The Integration Services service lets you use SQL Server Management Studio to *monitor running IS packages and to manage storage. to packages.*

## SQL Server Import and Export Wizard

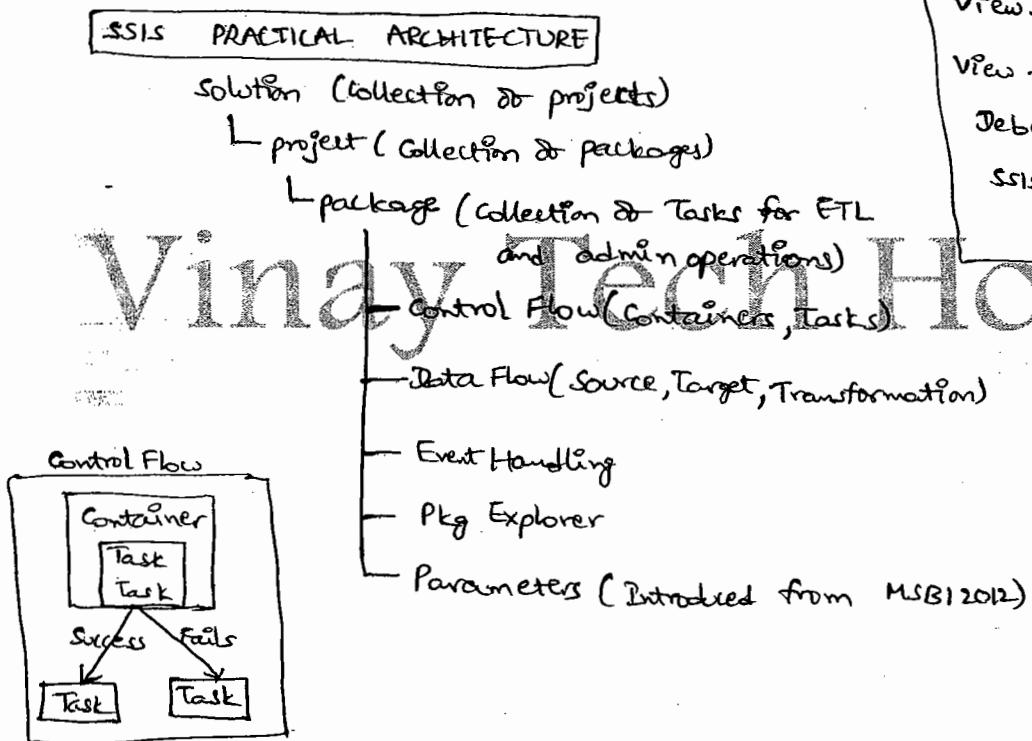
The SQL Server Import and Export Wizard can copy data to and from any data source for which a managed .NET Framework data provider or a native OLE DB provider is available. This wizard also offers the simplest method to create an Integration Services package that copies data from a source to a destination.

## Other tools, wizards, and command prompt utilities

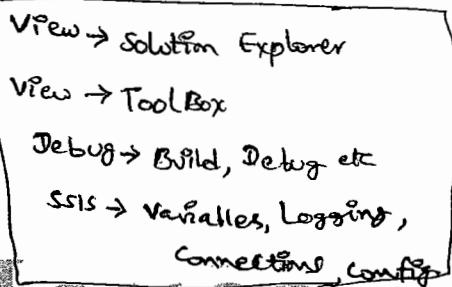
Integration Services includes additional tools, wizards, and command prompt utilities for running and managing Integration Services packages.

### SSIS PRACTICAL ARCHITECTURE

#### Diagram (Solution Hierarchy):



#### Important Navigations



#### Navigation:

- Solution Explorer Complete Hierarchy Navigation: Tools → options → projects & solution → always show solution

### SOLUTION

The starting point for SSIS is to create a solution and project.

A **solution** is a container in visual studio that holds one or many projects.

**Note:** you may have a solution that has your VB.NET application and all the SSIS packages that support that application. In this case, you would have two projects: one for VB and another for SSIS contained within the single solution.

**PROJECT**

A project in SSIS is a Container for one or more packages and related files.

**PACKAGE**

A core component of SSIS is the notion of a **package**. A package best parallels an executable program that you can write that contains workflow and business logic. Essentially, a package is a collection of tasks snapped together to execute in an orderly fashion. A package is also a unit of execution and development, much like a .NET developer creates programs or DLL files.

Precedence constraints are used to connect the tasks together and manage the order in which they execute, based on what happens in each task or based on rules defined by the package developer.

The package is brought together into a .dtsx file that is actually <sup>an XML structured file.</sup> with collections of properties. Just like other .NET projects, the file-based code is marked up using the development environment and can then be saved and deployed to a SQL Server.

```
<?xml version="1.0"?>
<DTS:Executable xmlns:DTS="www.microsoft.com/SqlServer/Dts"
DTS:refId="Package"
DTS:CreationDate="8/8/2011 12:53:33 AM"
DTS:CreationName="SSIS.Package.3"
DTS:CreatorComputerName="SSISServer"
DTS:CreatorName="SSIS_Brian"
DTS:DTSID="{FF3E3020-E008-4FF4-AC27-085AA1D21E88}"
DTS:ExecutableType="SSIS.Package.3"
DTS>LastModifiedProductVersion="11.0.1445.2"
DTS:LocaleID="1033"
DTS:ObjectName="Package1"
DTS:PackageType="5"
DTS:VersionGUID="{B2C67EE6-3510-4AD1-B11F-AF84C8DCA75C}">
<DTS:Property
DTS:Name="PackageFormatVersion">6</DTS:Property>
<DTS:Variables />
<DTS:Executables />
</DTS:Executable>
```

Navigation: first practice

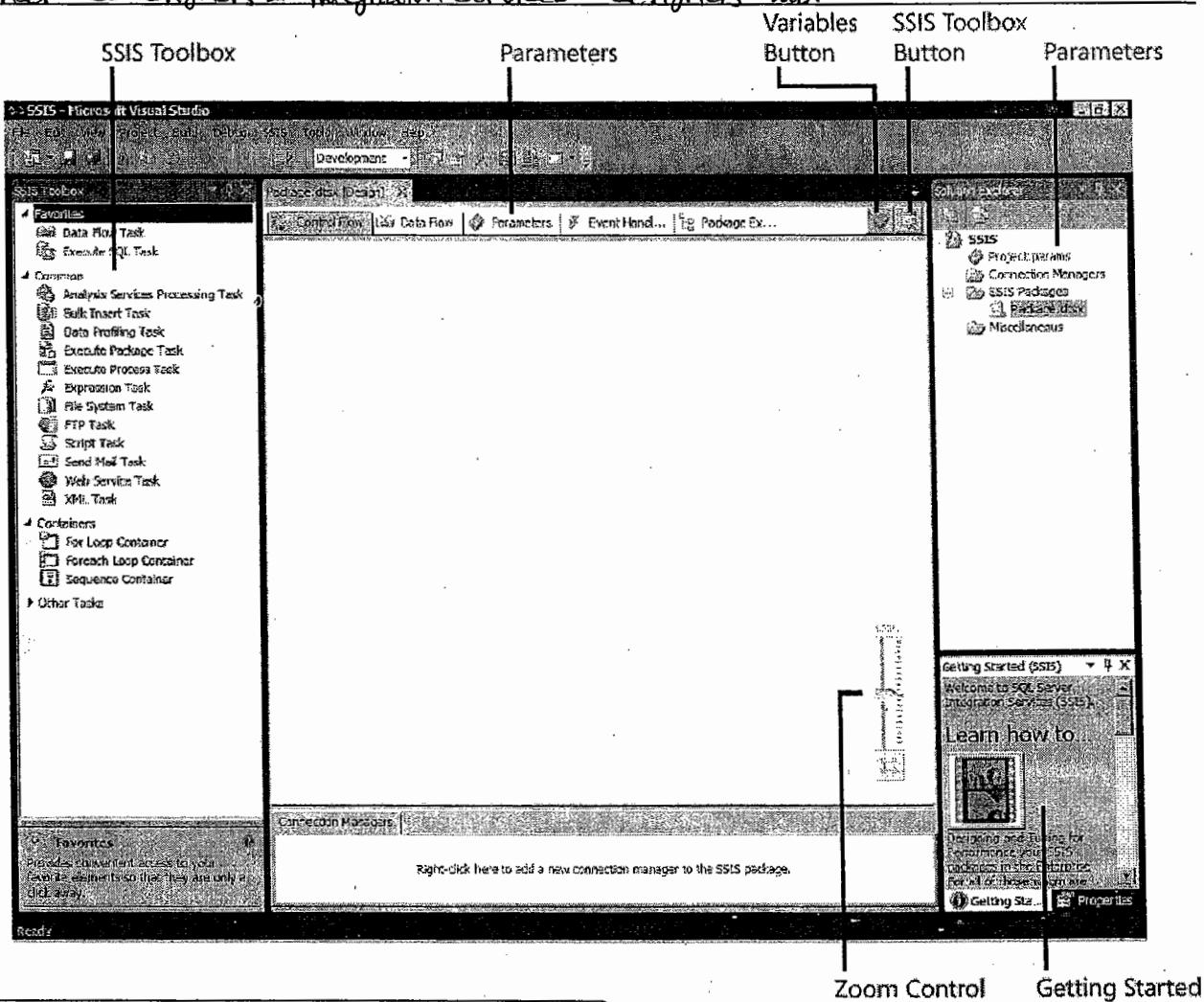
1. SSDT
2. file menu → new project → specify project name → solution name, directory to store → click OK.

## CONTROL FLOW

(Mandatory for every package)

The brain of a package is control flow, which orchestrates the order of execution for all its components. The components consist of tasks and containers and are controlled by *precedence constraints*.

To do this in SSDT, select Tools → Options, and then show precedence constraint labels under BI designers - integration services designers tab.



## APPLICATIONS & EXTENSIONS

- DTSX: An SSIS package, which uses its legacy extension from the early beta cycles of SQL Server 2005 when SSIS was still called DTS
- CONMGR: A connection manager that can be shared across any package in the entire project
- SLN: A solution file that contains one or more projects
- DTPROJ: An SSIS project file
- PARAMS: An SSIS project parameter file

## THE TOOLBOX

The Toolbox contains all the items that you can use in the particular tab's design pane at any given point in time. The Toolbox is organized into tabs such as favorites, common, containers and other tasks. These tabs can be collapsed and expanded for usability.

## The properties windows

You can use the Properties window to customize any item that you have selected in the Control Flow or Data Flow tabs.

## TASK GROUPING

A very nice usability feature in SSIS is the capability to group tasks logically in containers.

Note: Groups and containers are not the same. Groups are a usability feature to bring together components. Containers on the other hand allow you to pass properties into them.

**Navigation:** choose ~~set~~ ~~for~~ tasks → rc → Group or ungroup

## ANNOTATION

Annotations are a key part of any package, and a good developer won't fail to include them. An annotation is a comment that you ~~put~~ place in your package to help others & yourself understand what is happening <sup>in the package</sup>. To add an annotation, right-click where you want to place the comment, select Add Annotation and begin typing. It is a good idea to always add an annotation to your package that shows the title and version of your package.

**Navigation:** rc → control flow → add annotation.

Developer name : Mr. Bolego

Doc Ref : CLDW\_BDW.doc

Created date : 2014-12-29

Desc : This package moves data from party subject area to other subject area

## MISCELLANEOUS FOLDER

Goto project → rc → Add existing item.

If you copy any file that does not match the .PARAMS, .CONMGR or .DTSX extension, it will be placed in the Miscellaneous folder. This folder is used to hold any files that describe the installation of the package, such as Word documents or requirements documents.

## CONNECTION

They can connect to relational or Analysis Services databases, flat files, or other data sources.

There are 2 types of connection managers.

- a) Project level - across all projects we can use.
- b) Package level - within the package only we use.

## VARIABLES

**Variables** - In real time to hold a value, to pass a value at runtime variables are helpful.

They enable you to dynamically control the package at runtime, much like you do in any .NET language. There are two types of variables:

System variable : are those built into SSIS, such as a package name or the package's start time;

User variable : are created by the SSIS developer.

\* Variables stored under name spaces, system variables in system name space, user variable scope: in user name space.

They can also be set to be in the scope of a container, a task, or an event handler inside the package.

Eg:	Name	Scope	Datatype	Value
• ID		DataFlow Task	Int32	1000
• NAME		Package	String	VINAYTECH

## PARAMETERS

The Parameters tab enables you to create parameters for a package. These are different from variables in that they can easily be passed in from a DBA or a job.. You can find parameters in the list of tabs in SSIS. Parameters can be made secure by setting the Sensitive property to True, and they can be used to override nearly any property in SSIS. Some parameters can also be set to Required by setting the corresponding property to True, meaning the package won't run without passing in this parameter.

Parameters behave much like variables but with a few main exceptions. Parameters, like variables, can make a package dynamic. The largest difference between them is that parameters, can be set outside the package easily and can be designed as values that must be passed in for the package to start must like stored procedure s/p parameters. Parameters are new to SSIS in SQL Server 2012 and replace the capabilities of Configurations in previous releases of SQL Server.

## EVENT HANDLERS & HANDLING

It is called as exception handling in the prog. languages.

The Event Handlers tab enables you to create workflows to handle errors, warnings, or completion in tasks, containers, or packages. For example, if you want to trap any errors and have them e-mailed to you, you could create an OnError event handler that is scoped to the entire package and configure it to send a message out to an operator.

An executable can be a package, a Foreach Loop container, a For Loop container, a Sequence container, or a task.

EVENT	WHEN EVENT IS RAISED
OnError	When an error occurs
OnExecStatusChanged	When an executable's status changes
OnInformation	When an informational event is raised during the validation and execution of an executable
OnPostExecute	When an executable completes
OnPostValidate	When an executable's validation is complete
OnPreExecute	Before an executable runs
OnPreValidate	Before an executable's validation begins
OnProgress	When measurable progress has happened on an executable
OnQueryCancel	When a query has been instructed to cancel
OnTaskFailed	When a task fails
OnVariableValueChanged	When a variable is changed at runtime
OnWarning	When a warning occurs in your package



Example 1: send an email as soon as the package execution successful

## Event Handler tab

## Event Handler Package

## Event

## OnSuccess

Sendmail task and design it

Example 2: Delete the data in the table before Loading

Event handler tab

## Event handler

## DataFlows

## Event

## OnPreexecute

Execute Sql task with delete statement

## PACKAGE EXPLORER

This tab consolidates all the design panes into a single view, and lists all the tasks, connections, event handlers, containers, <sup>variables and transforms in your package</sup>. You can double-click any item here to configure it easily. You can also modify the properties for the item in the Properties window on the right after selecting the item you wish to modify. This tab is useful if you have a task that is throwing an error and you can't find it to remove or fix it.

## EXECUTING A PACKAGE & COLOURS OF A PACKAGE

There are

multiple ways to execute requires various colors.

- Press 'F5'
- Debug Menu → Start Debugging
- Clicking ➤ icon
- Solution Explorer → Package → RC → Execute Package
- By using Execute Package task.
- By using command line utilities (DTEXECUT etc.)
- By using Integration services → Package → Run package
- Execute package option in SQL Server catalog database (MSBI 2012) etc.

<u>Color</u>	<u>Meaning</u>
White	Ready for execution
Green	Success
Yellow	Running
Red	Fail
Grey	Disable
etc..	

**IN REAL TIME****GENERAL INPUT DOCUMENTS**

- CLIENT BUSINESS DOCUMENT  
(TALKS ABOUT OPERATIONS OF CLIENT)
- DATAWAREHOUSE NEED /SCOPE/PROCESSES
- HLD(HIGHLLEVEL DESIGN),LLD /SOURCE MAP/SOURCE TO TARGET/ BUSINESS MAF ETC...
- CHANGE REQUEST (CR),DEFECT TRACKS ETC..

**GENERAL OUT DOCUMENTS**

- UTILITY SCRIPT /SQL QUERY GENERATED
- UNIT TESTCASE AND RESULT DOC
- CONFIGURATION /PARAMETER FILES
- HLD CONVERTED LLDs
- ADHOC DOCS (CR,DEFECT,STATUS OF RUN ETC..)

LOW LEVEL DESIGN//MICRO DESIGN/SOURCE TO TARGET ETC...

---

THIS DOCUMENT IS USED FOR PKGS CREATION AND MANIPULATION.

GENERALLY DEVELOPER FRIENDLY DOCUMENT

IT CONTAINS

- A) SOURCE AND ITS STRUCTURE
- B) TARGET AND ITS STRUCTURE
- C) SOURCE TO TARGET BUSINESS RULES(TRANSFORMATION RULES)

DIRECT MOVE  
EG: EMPID LOADING TO PARTYID

HARD CODED  
EG: SET PARTYCODE AS 21000

TRANSFORMATION LOGIC  
EG:  
PIC SHOULD BE INCREMENTED BY 20%

## TASKS & BASIC OPERATIONS

A task can best be described as individual unit of work. Tasks provide functionality to your package, in much the same way that a method does in a programming language. You can develop your own tasks, but here are the current ETL tasks available to you out of the box:

**Analysis Services Execute DDL Task:** Executes a DDL Task in Analysis Services. For example, this can create, drop, or alter a cube.

**Analysis Services Processing Task:** This task processes a SQL Server Analysis Services cube, dimension, or mining model.

**Bulk Insert Task:** Loads data into a table by using the BULK INSERT SQL command.

**CDC Control Task:** Maintains and interacts with the change data capture (CDC) feature from SQL Server.

**Data Flow Task:** This very specialized task loads and transforms data into an OLE DB and ADO.NET destination.

**Data Mining Query Task:** Allows you to run predictive queries against your Analysis Services data-mining models.

**Data Profiling Task:** This exciting new task enables the examination of data; it replaces your ad hoc data profiling techniques.

**Execute Package Task:** Allows you to execute a package from within a package, making your SSIS packages modular.

**Execute Process Task:** Executes a program external to your package, such as one to split your extract file into many files before processing the individual files.

**Execute SQL Task:** Executes a SQL statement or stored procedure.

**Expression Task:** Sets a variable to an expression at runtime.

**File System Task:**

This task can handle directory operations such as creating, renaming, or deleting a directory. It can also manage file operations such as moving, copying, or deleting files.

**FTP Task:** Sends or receives files from an FTP site.

**Message Queue Task:** Sends or receives messages from a Microsoft Message Queue (MSMQ).

**Script Task:** This task enables you to perform .NET-based scripting in the Visual Studio Tools for Applications.

**Send Mail Task:** Sends a mail message through SMTP.

**Web Service Task:** Executes a method on a web service.

**WMI Data Reader Task:**

This task can run WQL queries against the Windows Management Instrumentation. This enables you to read the event log, get a list of applications that are installed, or determine hardware that is installed, to name a few examples.

**WMIEvent Watcher Task:** This task empowers SSIS to wait for and respond to certain WMI events that occur in the OS.

**XML Task:** Parses or processes an XML file. It can merge, split, or reformat an XML file.

## CONTAINERS

Containers are core units in the SSIS architecture for grouping tasks together logically into units of work. Besides providing visual consistency, containers enable you to define variables and event handlers (these are discussed in a moment) within the scope of the container, instead of the package.

### There are four types of containers in SSIS:

Task host container : Not a visible element that you'll find in the Toolbox, but rather an abstract concept like an interface.

Sequence Container :

Allows you to group tasks into logical subject areas. Within the development environment, you can then collapse or expand this container for usability.

For Loop Container : Loops through a series of tasks until a condition is met.

For each Loop Container : Loops through a series of files or records in a data set, and then executes the tasks in the container for each record in the collection.

## SSDT COMMON PROPERTIES

→ Common properties for maximum possible tasks.

\* There is a standard set of properties exposed in the design interface of the task.

### DelayValidation:

If set to true, SSIS will not validate any of the properties set in the task until runtime. This is useful if you are operating in a disconnected mode, and you want to enter a value for production that cannot be validated until the package is deployed, or you are dynamically setting the properties using expressions. The default value for this property is false.

### Description:

The description of what the instance of the task does. The default name for this is <task name>, or if you have multiple tasks of the same type, it would read <task name 1> (where the number 1 increments). This property does not have to be unique and is optional. If you do provide details here, it will display in the tooltip when hovering over the task object.

### Disable:

If set to true, the task is disabled and will not execute. This is helpful if you are testing a package and want to disable the execution of a task temporarily. This is the equivalent of commenting a task out temporarily. This property is set to false by default.

### ExecValueVariable:

Contains the name of the custom variable that will store the output of the task's execution. The default value of this property is <none>, which means the execution output is not stored. This variable provides for exciting workflow possibilities because it enables the task to expose information related to the results of the internal actions within the task.

### FailPackageOnFailure:

If set to true, the entire package will fail if the individual task fails. Typically, you want to control what happens to a package if a task fails with a custom error handler or Control Flow. Therefore, by default, this property is set to false.

\* **FailParentOnFailure:**

If set to true, the task's parent will fail if the individual task reports an error. The task's parent can be a package or container.

**ID:**

This is a read-only, automatically generated unique ID that is associated with an instance of a task. The ID is in GUID format and looks like this: {BK4FH3I-RDN3-I8RF-KU3F-JF83AFJRLS}.

**IsolationLevel:**

Specifies the isolation level of the transaction, if trans actions are enabled in the TransactionMode property. The values are Chaos, ReadCommitted, ReadUncommitted, RepeatableRead, Serializable, Unspecified, and Snapshot. The default value of this property is Serializable. These options correspond with standard SQL Server transaction types.

\* **LoggingMode:**

Specifies the type of logging that will be performed for this task. The values are UseParentSetting, Enabled, and Disabled. The default value of this property is UseParentSetting, which tells the task to use the logging mechanism for the package or container. Basic logging is turned on at the package level by default in SQL Server 2012.

**Name:**

The name associated with the task. The default name for this is <task name> or, if you have multiple tasks of the same type, <task name 1> (where the number 1 increments).

\* **TransactionOption:**

Specifies the transaction attribute for the task. Possible values are NotSupported, Supported, and Required. The default value of this property is Supported, which enables the option for you to use transactions in your task.

**Adding a New Package / Existing Package :** Solution explorer → SSIS Packages →

RC → add new package / existing package.

**Adding a New Project / Existing Project :** solution explorer → solution → RC → add new / existing project.

**Naming Conventions**

**Name the package.**

Name it something other than Package.dtsx. This matters later when you deploy the packages.

**Name packages with ETL verb extensions.**

Package Name\_Extract, Package Name\_Transformation, or Package Name\_Load. The **extension** **\_Process** seems to be explicit enough for those packages that don't fall into the other three categories.

**Provide a brief annotation about what the package does, where it gets inputs and outputs, and what to do if it fails.**

Annotations can include answers to the following questions:

Can it be rerun again? Is it part of a larger set of packages? Should it be restarted on checkpoints?

**Add short descriptive words to SSIS tasks and components, but don't alter the name altogether.** For example, change an Execute SQL Task to Execute SQL Task to Retrieve Invoice ID. Use the Description field on the object to provide the detailed information.

Ex : PKG\_PARTY SUBJECT\_AREA\_LOAD

Vinay Tech House # 505, Annapurna Block, Aditya Enclave, Ameerpet, Hyd. Ph: 04066638869/9573168449  
JF\_PARTY\_TABLE\_LOAD

ESQT\_PARTY\_TABLE\_EXTRACT.

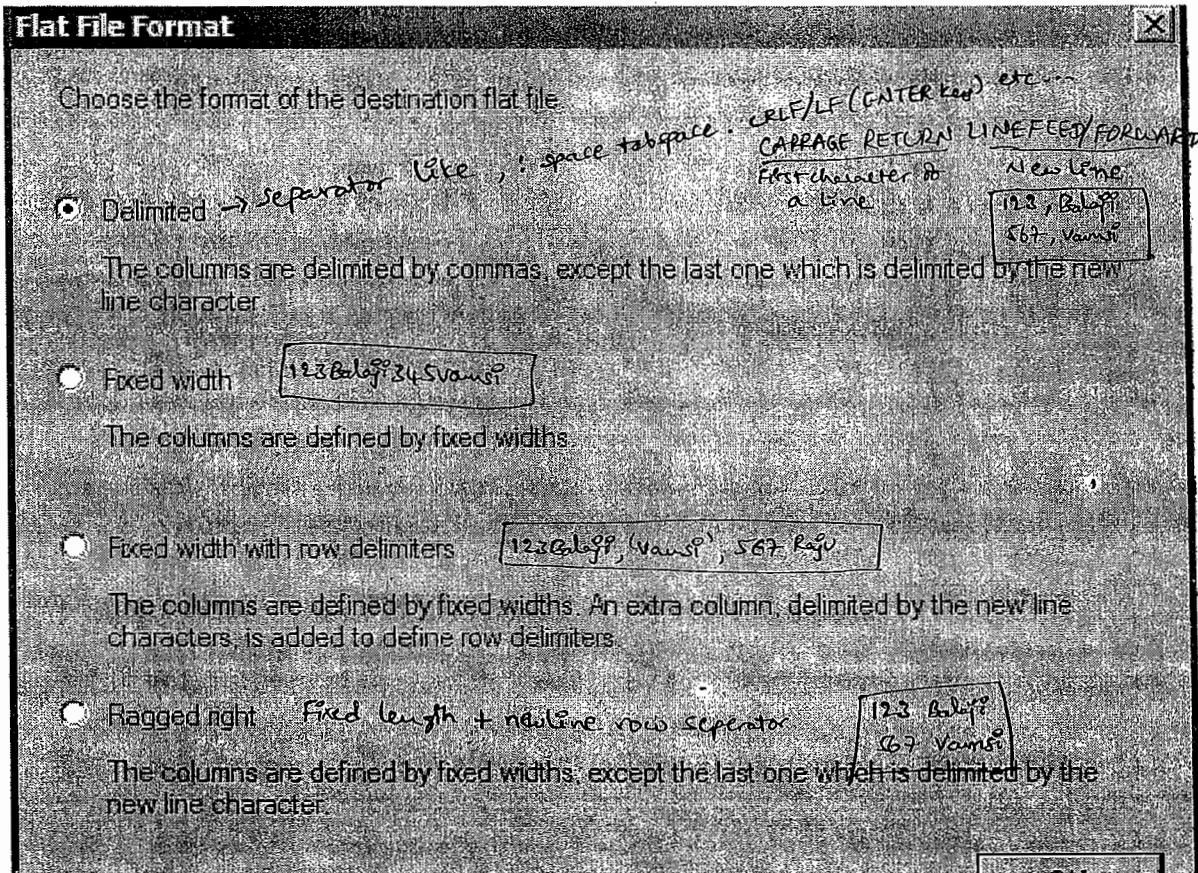
1. Stage area is an intermediate store area, In my current browser I have two stage areas.  
a. File Stage area b. Database stage area.

## MSBI 2012 (SQL Server Integration Services)

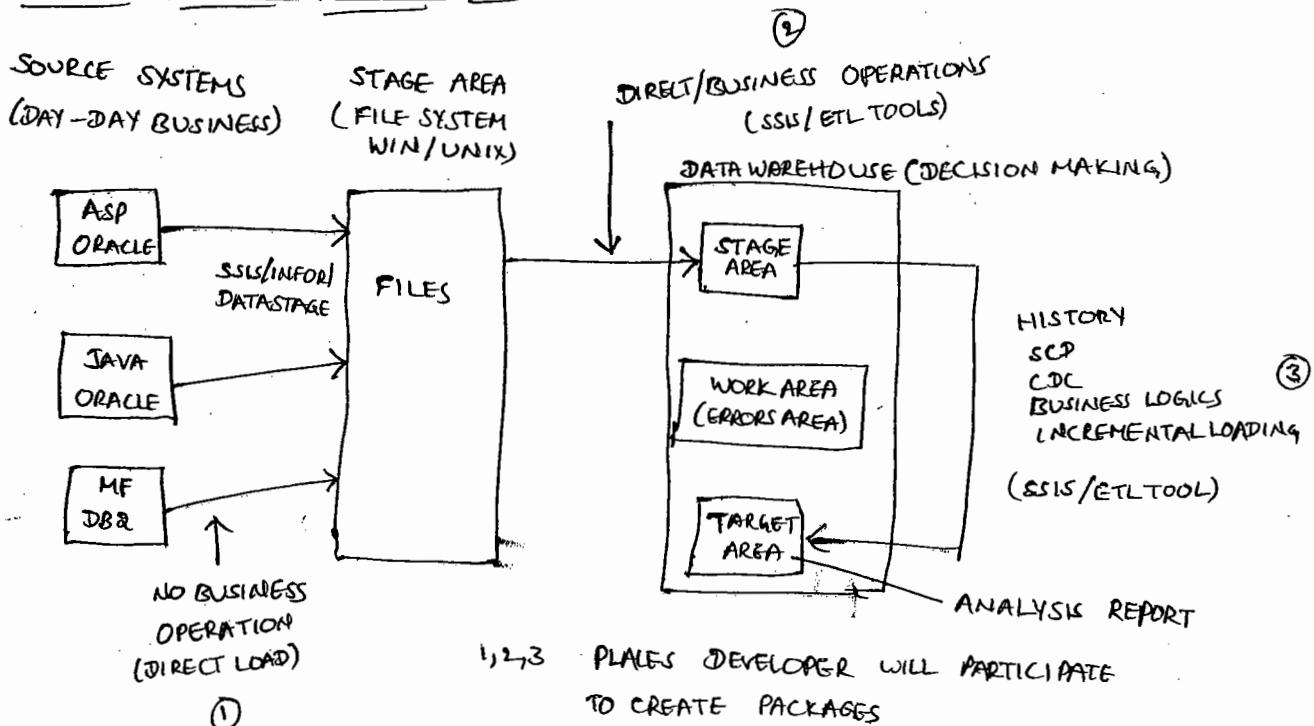
22

### FLAT FILE & TYPES

1. Universal acceptable file, platform independent.
2. In real time companies use files for intermediate storage.



### SSIS Realtime project flow:



E-AMS → Active Regular Insurance customers information.

Vinay  
TechHouse

# 505, Annapurna Block, Aditya Enclave, Ameerpet, Hyd. Ph: 04066638869/9573168449

① What is source system and explain types of it?

Simple day-day operation System is source system. In my company we have the below

## DATA LOADING

a) Without Transformation Logic - (Direct load) → In real time stage to stage (or source to stage) is direct load.  
There are three ways

- i) Bulk insert task
- ii) Import Export wizard
- iii) Dataflow task

b) With Transformation Logic - (Business logic load) → By using dataflow only we do it.

NOTE: Generally in realtime stage 2 target load is transformation load.

**IMPORT EXPORT WIZARD** → Loading unloading through graphical.

The Import and Export Wizard is the easiest method to move data from sources like Oracle, DB2, SQL Server, Excel, and text files to nearly any destination, and it is available across all versions of SQL Server — even those that don't include SSIS. This wizard uses SSIS as a framework and can optionally save a package as its output prior to executing.

LOADING → IMPORTING → POPULATING

UNLOAD → EXPORTING → UNPOPULATING

\* Available in 3 places.

1. START → PROGRAMS → SQL Server 2012

2. SSDT → SSIS Packages → LC → --

3. SSMS → DATABASE ENGINE → DATABASE → LC → IMPORT EXPORT

PROCESS: Source servername, UID, PWD, table/filename  
Target " " "

Navigation: Move target table from DB\_1 to DB\_2

1. Open wizard in any of above navigation

2. Source servername (Instance name)  
DB name : DB\_1

Target - Server name (Instance name)

DB name : DB\_2

Object to copy choose Test

NEXT → NEXT → FINISH.

**Exercises: a) Load data from a table to FlatFile ragged right format**

## BULK INSERT TASK

The Bulk Insert Task enables you to insert data from a text or flatfile into SQL server database table in the same high octane manner as using a bulk insert statement. In fact, the task is basically just a wizard to store the information needed to create and execute a bulk copying command at runtime (similar to BCP from a command line). The downside of the Bulk Insert Task is its strict data format, and it precludes being able to work with data in a Data Flow within one action. This can be seen as a disadvantage in that it does not allow any transformations to occur to the data in flight, but not all ETL processes are efficiently modified in the Data Flow.

**Use File:** This uses the BCP format (.fmt) file.

**Specify:** This enables you to select the file delimiters. The available delimiters are New Line ({CR}{LF}), Carriage

Return ({CR}), Line Feed ({LF}), Semicolon (;), Comma (,), Tab, or Vertical Bar (|). Note that the defaults are for the row to be {CR}{LF} delimited, and the column tab-delimited.

**Code page:** You can specify the code page for the source file. You will rarely want to change the code page from RAW,

which is the default. Using RAW is the fastest data-loading option because no code page conversion takes place.

**OEM:** You should use this when copying from one SQL Server to another.

**ACP:** This converts non-Unicode data to the ANSI code page of the SQL Server you are loading the data into or you can specify a specific code page mapping.

**DataFileType:** Specifies the type of the source file. Options here include char, native, wide char, and wide native. Generally, files you receive will be the default option, char, but in some cases, you may see a file with a native format.

→ You can also use the Options tab to specify the first and last row to copy if you want only a sampling of the rows. This is commonly used to set the first row to two (2) when you want to skip a header row.

→ The BatchSize option indicates how many records will be written to SQL Server before committing the batch. A BatchSize of 0 (the default) means that all the records will be written to SQL Server in a single batch. If you have more than 100,000 records, then you may want to adjust this setting to 50,000 or another number based on how many you want to commit at a time.

The Options dropdown contains five options that you can enable/disable:

✓ **Check Constraints:** This option checks table and column constraints before committing the record. It is the only option enabled by default.

**Keep Nulls:** By selecting this option, the Bulk Insert Task will replace any empty columns in the source file with NULLs in SQL Server.

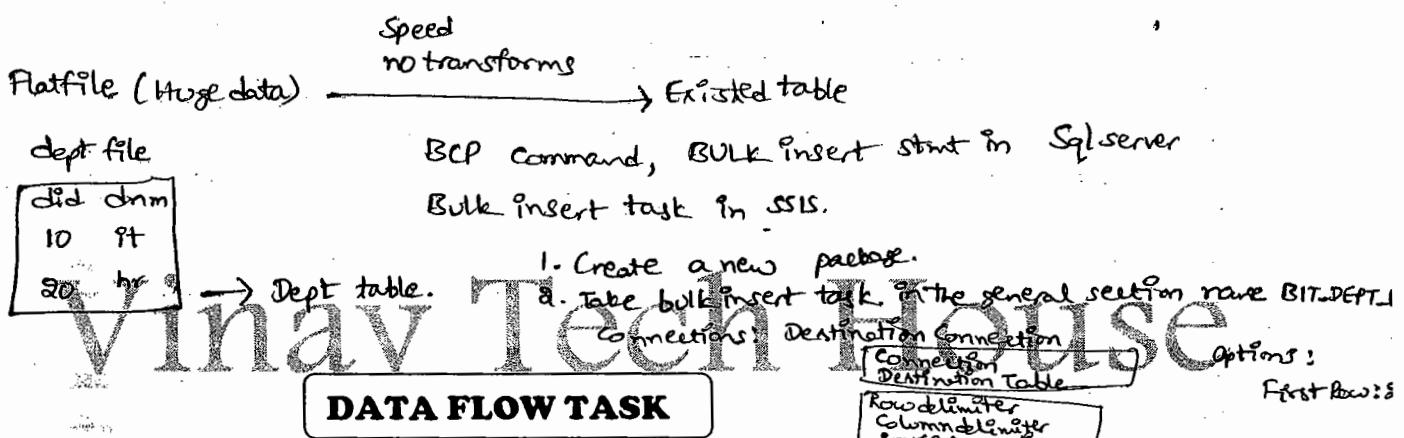
**Enable Identity Insert:** Enable this option if your destination table has an identity column into which you're inserting. Otherwise, you will receive an error.

**✓Table Lock:** This option creates a SQL Server lock on the target table, preventing inserts and updates other than the records you are inserting. This option speeds up your process but may cause a production outage, as others are blocked from modifying the table. If you check this option, SSIS will not have to compete for locks to insert massive amounts of data into the target table. Set this option only if you're certain that no other process will be competing with your task for table access.

**Fire Triggers:** By default, the Bulk Insert Task ignores triggers for maximum speed. When you check this option, the task will no longer ignore triggers and will instead fire the insert triggers for the table into which you're inserting.

#### Real time usage:

When we move data from stage files to stage or target database bulk insert task is helpful.



The core strength of SSIS is its capability to extract data into the server's memory, transform it, and write it out to an alternative destination. **If the Control Flow is the brains of SSIS, then the Data Flow would be its heart.** The in-memory architecture is what helps SSIS scale and what makes SSIS run faster than staging data and running stored procedures.

Much of this ETL process is done in memory which is what gives SSIS its speed. It is much faster to apply business rules to your data in memory using a transformation than to have to constantly update a staging table. Because of this, though, your SSIS server will potentially need a large amount of memory, depending on the size of the file you are processing.

Data flows out of a source in memory buffers that are 10 megabytes in size or 10,000 rows (whichever comes first) by default.

#### SOURCES

→ A datafeed (or) data input (or) data producer.

A source is a component that you add to the dataflow design surface to specify the location of source data that will send data to components down stream.  
SSIS provides eight out-of-the-box sources:

1. **OLE DB Source:** Connects to nearly any OLE DB data source, such as sql server, oracle, axes, DB2 et

2. **Excel Source:** Specializes in receiving data from Excel spreadsheets.
3. **Flat File Source:** Connects to a delimited (or) fixed width.
4. **Raw File Source:** Produces a specialized binary file for data that is in transit; it is especially quick to read by SSIS.
5. **XML Source:** Retrieves data from an XML document. This source does not use a Connection Manager to configure it.
6. **ADO.NET Source:** This source is just like the OLE DB Source but only for ADO.NET based sources. The internal implementation uses an ADO.NET DataReader as the source. The ADO.NET connection is much like the one you see in the .NET Framework when hand-coding a connection and retrieval from a database.
7. **CDC Source:** Reads data out of a table that has change data capture (CDC) enabled. Used to retrieve only rows that have changed over a duration of time.
8. **ODBC Source:** Reads data out of table by using an ODBC provider instead of OLE DB. When you are given the choice between OLE DB and ODBC, it is still recommended in SSIS packages that you use OLE DB.

#### **SOURCE ASSISTANT AND DESTINATION ASSISTANT → Introduced in 2012 to reduce complexity & configuring source and destination connections multiple times.**

The Source Assistant and Destination Assistant are two components designed to remove the complexity of configuring a source or a destination in the Data Flow. The components determine what drivers you have installed and show you only the applicable drivers. It also simplifies the selection of a valid connection manager based on the database platform you select that you wish to connect to.

In the Source Assistant or Destination Assistant (the Source Assistant is shown in Figure 5-1), only the data providers that you have installed are actually shown. Once you select how you want to connect, you'll see a list of Connection Managers on the right that you can use to connect to your selected source. You can also create a new Connection Manager from the same area on the right. If you uncheck the "Show only installed source types" option, you'll see other providers like DB2 or Oracle for which you may not have the right software installed.

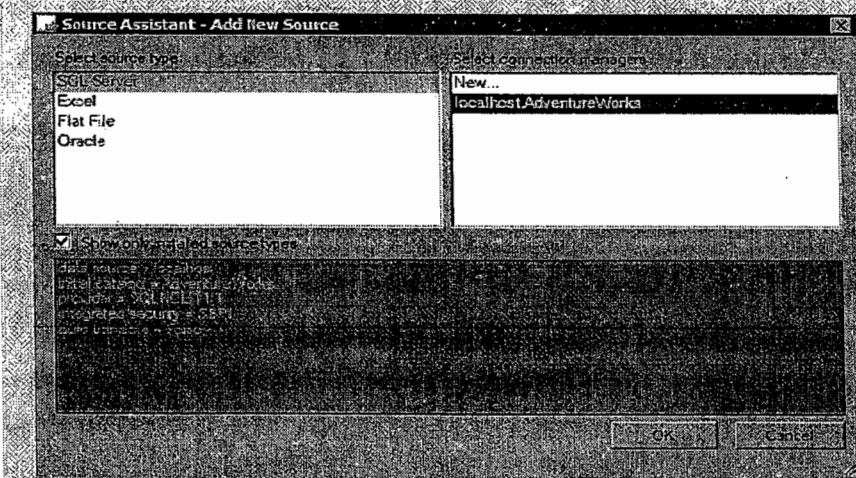


FIGURE 5-1

**DESTINATIONS**

(ii) Consumers.

Inside the Data Flow, destinations consume the data after the data pipe leaves the last transformation components.

The following destinations are available to you in SSIS:

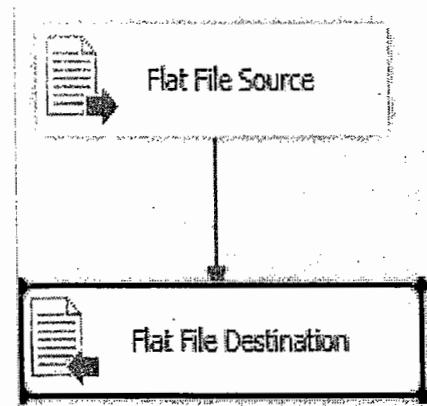
1. **ADO.NET Destination:** Exposes data to other external processes, such as a .NET application.
2. **Data Mining Model Training:** Trains an Analysis Services mining model by passing data from the Data Flow to the destination.
3. **Data Reader Destination:** Allows the ADO.NET DataReader interface to consume data, similar to the ADO.NET Destination.
4. **Dimension Processing:** Loads and processes an Analysis Services dimension. It can perform a full, update, or incremental refresh of the dimension.
5. **Excel Destination:** Outputs data from the Data Flow to an Excel spreadsheet.
6. **Flat File Destination:** Enables you to write data to a comma-delimited or fixed-width file.
7. **ODBC Destination:** Outputs data to an ODBC data connection like SQL Server, DB2, or Oracle.
8. **OLE DB Destination:** Outputs data to an OLE DB data connection like SQL Server, Oracle, or Access.
9. **Partition Processing:** Enables you to perform incremental, full, or update processing of an Analysis Services partition.
10. **Raw File Destination:** Outputs data in a binary format that can be used later as a Raw File Source. It's usually used as an intermediate persistence mechanism.
11. **Recordset Destination:** Writes the records to an ADO record set. Once written, to an object variable, it can be looped over a variety of ways in SSIS like a Script Task or a Foreach Loop Container.
12. **SQL Server Compact Edition Destination:** Inserts data into a SQL Server running the Compact Edition of the product on a mobile device or PC.
13. **SQL Server Destination:** The destination that you use to write data to SQL Server. This destination has many limitations, such as the ability to only write to the SQL Server where the SSIS package is executing.

**Data Direct Loading in Data Flow: BUSINESS SCENARIO:**

FILENAME	FILE COLUMNS	FILENAME	COLUMNNAME	TRANSFORMAT ION RULE	KEYVAL UE
PARTY_TGT	PARTY_ID	PARTY_S RC	PARTYID	DIRECT MOVE	Y
PARTY_TGT	PARTY_NAME	PARTY_S RC	PARTYNAME	DIRECT MOVE	
PARTY_TGT	PARTY_LOC	PARTY_S RC	PARTYLOC	DIRECT MOVE	
PARTY_TGT	PARTY_INCOME	PARTY_S RC	PARTYINCOME	DIRECT MOVE	

To work with flatfile the below things we should identified first.

### Flat File TO Flat File Load



#### Navigation & Points

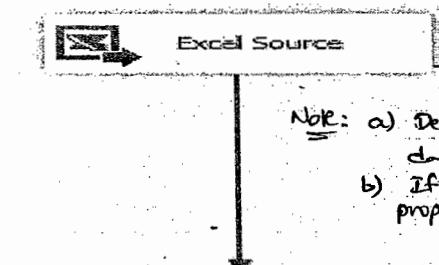
→ Use file source PARTY\_SRC from input folder.

#### flat file settings:

- PATH (Machine name/directory/filename.ext).
- Column separator
- row "
- skipping rows
- Header rows
- Text qualifier.

NOTE: Use INPUT, OUTPUT folders and DB-MSB1 database for the operation.

### Excel to Excel Data Loading



#### Navigation & Points

→ Use input folder PARTY\_SRC to output folder PARTY\_TGT

rc → Edit → Excel connection mgr → Browse PARTY\_SRC.xls file  
Name of the worksheet: SRC

- Note:
- Destination worksheet should have valid column names. (Otherwise total data loaded to single column).
  - If there are issues with 64-bit, please goto project properties → configuration properties → debugging → Run 64-bit runtime → False.

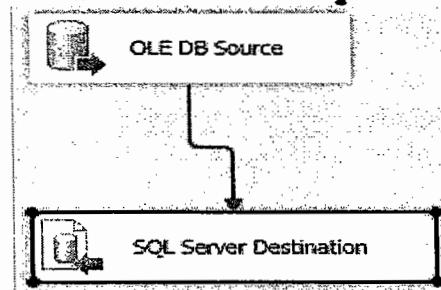
rc → Excel connection mgr → Browse PARTY\_TGT.xls file

Name of the worksheet: TGT

Mappings: Connect / map columns → OK

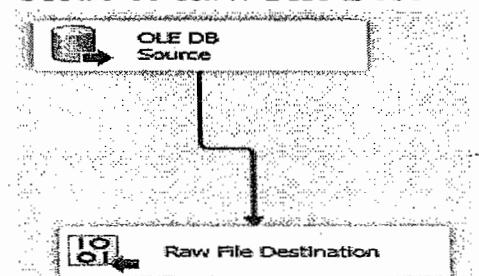
\*\*Exercises (Practice at home):

### Oledb Source to SqlServer Dest



#### Navigation & Points

### Oledb to Raw File Dest



#### Navigation & Points

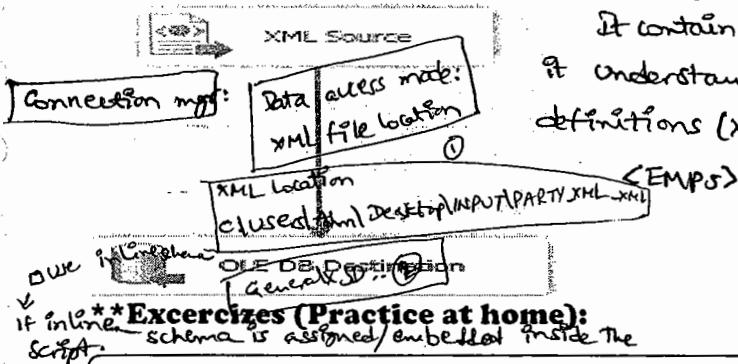
rc → edit → Access mode: FileName

FileName: Browse to o/p folder and specify a name.

Write option: Create Always

Columns: select required columns → OK

Note: There is no mapping section for raw destination.

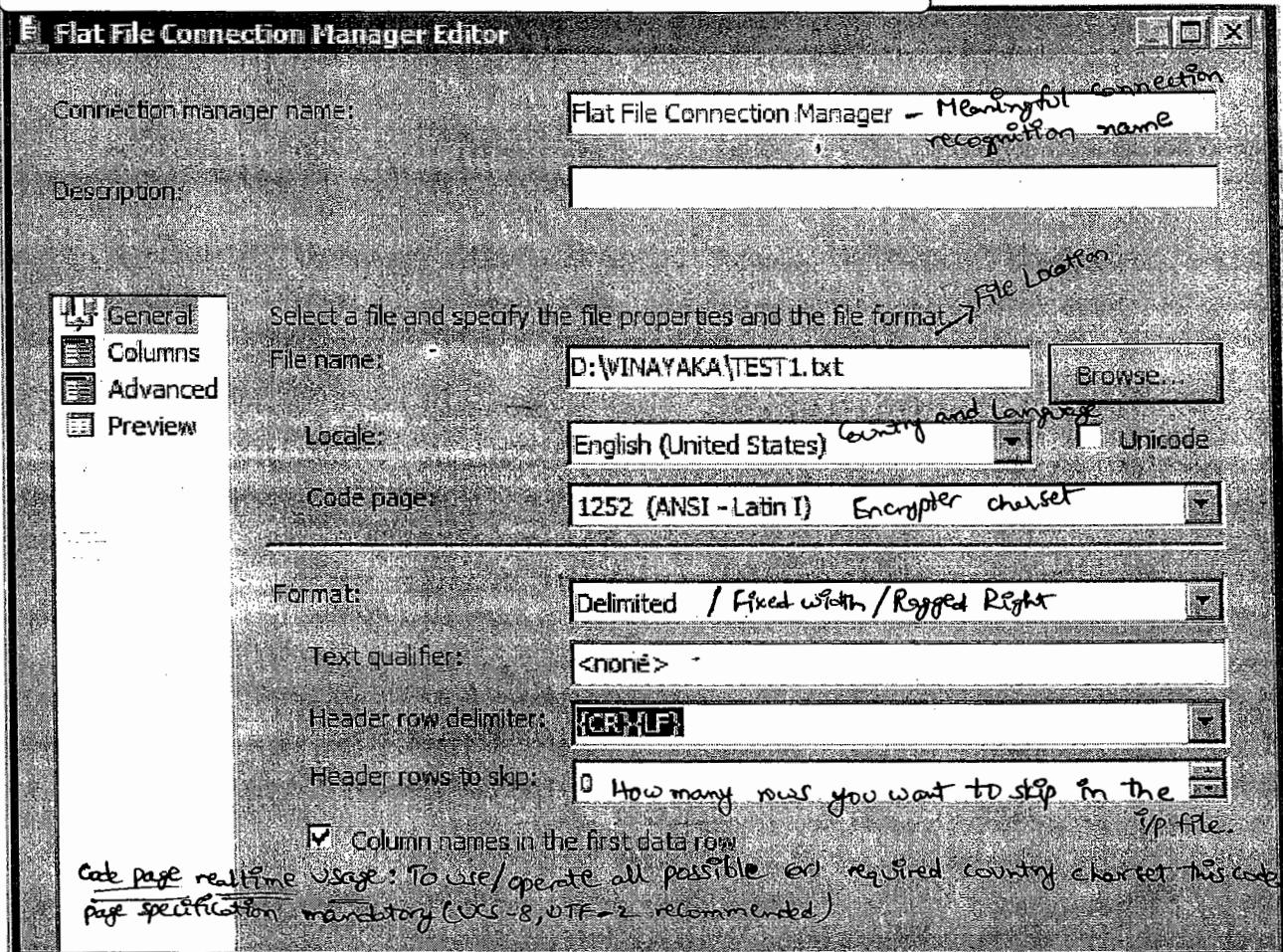
**XML Source to Oledb Dest**

It is an universal file format for better data processing. It contain tags, every tag should close properly. To make it understandable to the system we should pass schema definitions (XSD).

```

<EMP>
  <EMP1>
    <EID>1 </EID>
    <ENAME>X </ENAME>
  </EMP1>
  <EMP1>
    <EID>2 </EID>
    <ENAME>Y </ENAME>
  </EMP1>
</EMP>

```

**FLAT FILE SOURCE ADDITIONAL PROPERTIES**

- Retain null values from the source as null values in the dataflow. → When we check this option system consider NULL as NULL. Otherwise system treats NULL as '0' or 'space'.

Text qualifier: How do you want to qualify the text (simply recognition of data through 80 chars).

Text Qualifier

INPUT

" ", "vina"

OUTPUT

' ', 'Vinay' Text qualifier : 'single quote'

Column section: It shows row and columns separator.

Advance: To add remove column it will change properties of the columns such as type, Length etc this is helpful.

Preview: To see some sample records.

Show advanced editor: Once the connection mgr has been set to modify the column settings advanced editor is helpful.

Navigation: Flatfile → rc → show advanced editor → i/p o/p prop. → ~~description~~ source d.

Columns: Display the XML in Internal columns.

\* Error o/p: This is considered as error treatment option.

\* From the i/p if you are getting an error/truncation what you want to do? It talks about.

There are 3 options. a) fail component - for source error/truncation it fails.  
b) redirect row - for source error/truncation redirect into a destination.

**Vinay TechHouse**

#### FastParse Option

By default, SSIS issues a contract between the Flat File Source and a Data Flow. It states that the source component must validate any numeric or date column. For example, if you have a flat file in which a given column is set to a four-byte integer, every row must first go through a short validation routine to ensure that it is truly an integer and that no character data has passed through. On date columns, a quick check is done to ensure that every date is indeed a valid in-range date.

This validation is fast but it does require approximately 20 to 30 percent more time to validate that contract. To set the FastParse property, go into the Data Flow Task for which you're using a Flat File Source. Right-click the Flat File Source and select Show Advanced Editor. From there, select the Input and Output Properties tab, and choose any number or date column under Flat File Output → Output Columns tree. In the right pane, change the FastParse property to True.

### OLEDB SOURCE ADDITIONAL PROPERTIES

Data access mode: Data retrieval options from source.

Table or view

Table or view

Table name or view name variable

SQL command

SQL command from variable

Table (or) view: If you know the table (or) view name is fixed.

Table name (or) view name variable: Till runtime if you don't know the table (or) view name, passing through a variable.

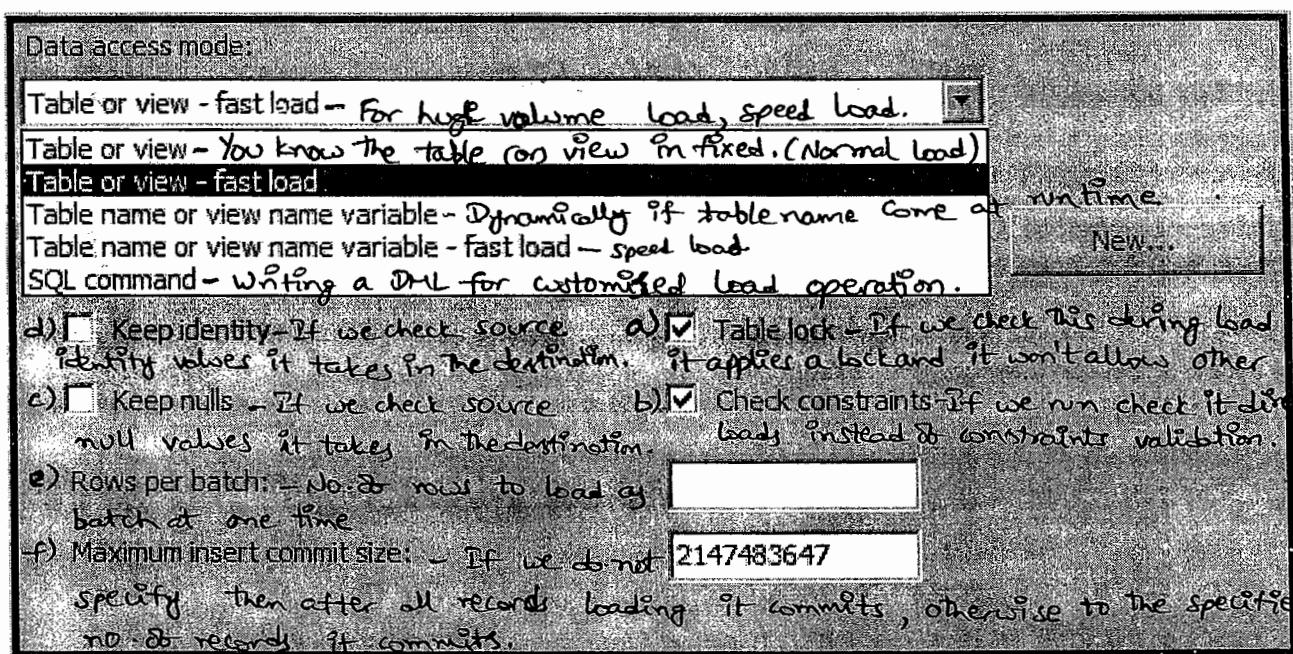
\* SQL command: for customized rows and columns.

Ex: 1. Single table required columns 2. from multiple tables getting required cols by joining the tables.

NOTE: Here we can write only select statement.

SQL command from variable: The command is changing dynamically, passing through variable this is recommended.

## OLEDB DESTINATION ADDITIONAL PROPERTIES



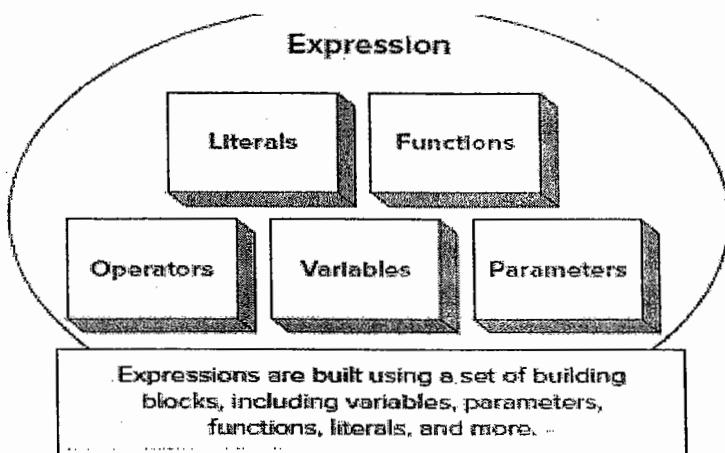
### Table/view

- Normal load
- Record by record load
- Recommended in all situations in realtime
- No additional considerations (or) options.

### Table/view - fast load

- Bulk load
- set db records at one time load (Batch load)
- Recommended in the below situations:
  - a) No/less constraints we have
  - b) on clustered index table.
- Additional settings we need to consider
  - a) b) c) d) e) f)

## DYNAMIC PACKAGE OBJECTS (EXP/VAR /PARM /LITERAL /FUNCTIONS)



SSIS includes multiple objects that can be used to create dynamic packages. SSIS can dynamically set a property in a task or component using an expression, which can be built using a set of building blocks including variables, parameters, functions, literals and more. When the package is run, the expression is evaluated and used in the property as each task or component is accessed.

One needs to decide when it is best to use an expression. A few pointers that I would like to provide.

- Use expressions where they are small
- Do not make the expression overtly complicated. It makes management difficult.
- Avoid too many nested conditions in expressions.

Expressions can be used at various places:

1. Precedence constraint - Should evaluate to TRUE or FALSE
2. Setting up variable value - Should be of the variable data type
3. For Loop
4. Setting up properties for various objects dynamically especially Connection Strings for various Connection Managers
5. Derived Column Transform (most often used here)
6. Conditional Split

## Parameter Overview

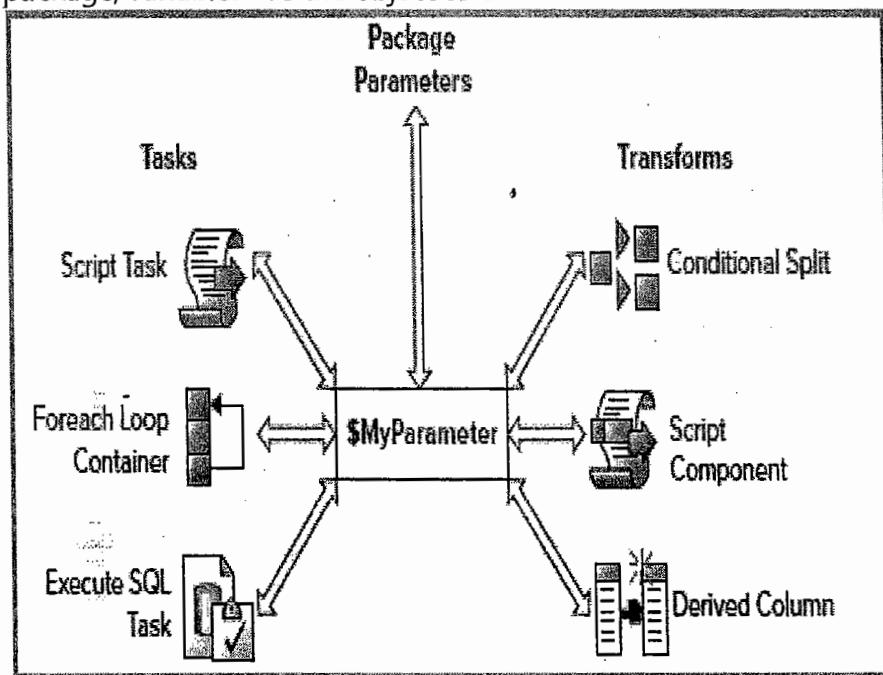
Parameters are a new feature introduced in SQL Server 2012. While similar to a variable in that a parameter can store information and be used in an expression, it has a few different properties and uses that you will want to understand. As demonstrated in Figure 6-2, parameters are set externally. The parameter can then be used in an expression to affect different properties in tasks and components.

SSIS uses two types of parameters: a) Project parameter b) Package parameter.

Project parameters are created at the project level and can be used in all packages that are included in that

project. On the other hand, package parameters are created at the package level and can be used only in that package. Project parameters are best used for values that should be shared among packages, such as e-mail addresses for error messages. Package parameters are best used for values specific to that package, such as directory locations.

When using the project deployment model, parameters are the best choices to replace package configurations to create a dynamic and more flexible SSIS solution. Using the Required property of the parameter, you can also necessitate the caller of the package to pass in a value for the parameter. If you want to set the value of something from outside the package, either required or not, parameters are the object to use. If you want to create or store values only within a package, variables are the object to use.



House

Mathematical Functions

FUNCTION	Result	Data Type
ROUND( 2.23 , 2 ) <i>~ Rounding to nearest integer</i>	2	Integer
ROUND( 2.53 , 2 )	3	Integer
CEILING(2.23) <i>- upper integer value</i>	3	Numeric
FLOOR(2.9) <i>- floor integer value</i>	2	Numeric
ABS(3.2) <i>1. Converts -ve to +ve 2. Rounds to the nearest integer.</i>	3	Numeric
ABS(3.9)	4	Numeric
ABS(-3.2)	3	Numeric

ABS(-3.9)	4	Numeric
<b>String Functions</b>		
FUNCTION	Result	DataType
LOWER( "ABD" )	abc	String
UPPER( "abd" )	ABD	String
LTRIM( " abc " )	"abc "	String
RTRIM( " abc " )	" abc"	String
TRIM( " abc " )	"abc"	String
REVERSE( "abc" )	"cba"	String
RIGHT( abc, 2 )	"bc"	String
* SUBSTRING( "abcde", 2, 3 ) Takes a part of a string in the specified length.	"bcd"	String position to the
REPLICATE("a",3)	"aaa"	String
FINDSTRING("abcd", "ab", 1)	"ab"	String
REPLACE("Apple", "pp", "app")	"An apple"	String
<b>Date/Time Functions</b>		
FUNCTION	Result	DataType
DATEADD( "mm", 1, (DT_DATE)"12/24/2009" ) for the corresponding date 1 month adding	1/24/2010 0:00	DateTime
DATEADD( "dd", -1, (DT_DATE)"12/24/2009" )	12/23/2009 0:00	DateTime
DATEADD( "yy", 1, (DT_DATE)"12/24/2009" )	12/24/2010 0:00	DateTime
DATEDIFF( "dd", (DT_DATE)"12/24/2009", DT_DATE)"01/26/2010" )	33	Integer
DATEDIFF( "mm", (DT_DATE)"12/24/2009", (DT_DATE)"01/26/2010" )	1	Integer
DATEDIFF( "yy", (DT_DATE)"12/24/2009",	1	Integer

(DT_DATE)"01/26/2010" )		
DATEPART( "dd", (DT_DATE)"12/24/2009" )	24	Integer
DATEPART( "mm", (DT_DATE)"12/24/2009" )	12	Integer
DATEPART( "yy", (DT_DATE)"12/24/2009" )	2009	Integer
DAY( (DT_DATE)"12/24/2009" )	24	Integer
MONTH( (DT_DATE)"12/24/2009" )	12	Integer
YEAR( (DT_DATE)"12/24/2009" )	2009	Integer

**NULL Functions**

FUNCTION	Result	DataType
ISNULL( <<expression>> )	TRUE/FALSE	Boolean
NULL(DT_DATE)	NULL	DateTime

**CASTING OPERATOR**

DT\_STR(&lt;&lt;length&gt;&gt;, &lt;&lt;code\_page&gt;&gt;)

length — Final string length

code\_page — Unicode character set

DT\_WSTR(&lt;&lt;length&gt;&gt;)

length — Final string length

DT\_NUMERIC(&lt;&lt;precision&gt;&gt;, &lt;&lt;scale&gt;&gt;)

precision — Max number of digits scale —

Number of digits after decimal

DT\_DECIMAL(&lt;&lt;scale&gt;&gt;)

scale — Number of digits after decimal

DT\_BYTES(&lt;&lt;length&gt;&gt;)

length — Number of final bytes

DT\_TEXT(&lt;&lt;code\_page&gt;&gt;)

code\_page — Unicode character set

**EXPRESSION OPERATOR****DESCRIPTION**

||

Logical OR operation - Any condition to be met

&amp;&amp;

Logical AND operation - All conditions to be met

==

Comparison of two expressions to determine equivalency

!=

Comparison of two expressions to determine inequality

?:

Conditional operator

&lt;condition&gt; ? &lt;success stat&gt; : &lt;failure stat&gt;

Ex: ISNULL(SAL) ? 9000 : SAL

PARTY LOC == "HYD" ? "CORRECT" : "WRONG"

# 505, Annapurna Block, Aditya Encalve, Ameerpet, Hyd. Ph: 04066638869/9573168449

EXPRESSION FUNCTION	DESCRIPTION	C# EQUIVALENT
---------------------	-------------	---------------

POWER () Raise numeric to a power

Pow ()

LOWER () Convert to lowercase

ToLower ()

GETDATE () Return current date

Now ()

EXPRESSION FUNCTION	DESCRIPTION	DIFFERENCE
---------------------	-------------	------------

DATEPART () Parses date part from a date

Requires quotes around the date part

ISNULL () Tests an expression for NULL

Doesn't allow a default value

SUFFIX	DESCRIPTION	EXAMPLE
--------	-------------	---------

L or l Indicates that the numeric literal should be interpreted as the long version of either the DT\_I8 or DT\_R8 value types depending upon whether a decimal is present

3000000000L ↳ DT\_I8

3.14159265L ↳ DT\_R8

U or u Indicates that the numeric literal should represent the unsigned data type

3000000000UL ↳ DT\_UI8

SUFFIX	DESCRIPTION	EXAMPLE
--------	-------------	---------

F or f Indicates that the numeric literal represents a float value

100.55f ↳ DT\_R4

E or e Indicates that the numeric literal represents scientific notation

6.626 × 10 -34 J/s ↳ 6.626E-34F ↳ DT\_R8

Note: Expects at least one digit scientific notation followed by float or long suffix

6.626E won't work. If you don't have a digit, then format as follows:

6.626E+0L or 6.626E+0f

SUFFIX	DESCRIPTION	EXAMPLE
--------	-------------	---------

\n New Line or

"Print this on one line\nThis on another"

Carriage

Print this on one line

Feed Line Return

This on another

\t Tab character

"Print\twith\ttab\tseparation"

Print with tab separation

\" Double-quotation mark character

"\\"Hey! \"

"Hey! "



\ Backslash

"c:\\myfile.txt"

c:\\myfile.txt

### **Equivalence Operator**

`@MyBooleanValue = True`

### **String Concatenation**

"The Server [" + LOWER( @[System::MachineName]) + "] is running this package"

### **Line Continuation**

```
"My Line breaks here\nAnd then here\n; "
My Line breaks here
And then here
; )
```

### **Literals**

Literals are hard coded information that you must provide when building expressions. SSIS expressions have three types of literals: **numeric, string, and Boolean**.

#### **Numeric Literals**

A numeric literal is simply a fixed number.

#### **String Literals**

When building strings, there are times when you need to supply special characters in them

#### **Boolean Literals**

The Boolean literals of True and False don't have to be capitalized, nor are they case sensitive.

Boolean expressions are shortcut versions of the logical operators.

`@[System::OfflineMode] = True ? 1 : 0` (Not Recommended) `@[System::OfflineMode] = False`

### **NULLs and Variables**

VARIABLE DATA TYPE	DEFAULT VALUE
Boolean	False
Byte	0
Char	0
DateTime	12/30/1899
DBNull	(Can't test in an expression)
Double	0
Int16	0
Int32	0
Int64	0
Object	(Can't test in an expression)
SByte	0
Single	0
String	'' (empty string)
UInt32	0
UInt64	0

### **Some Expressions**

Get today's date in the format DD/MM/YYYY

```
1.RIGHT("0" + (DT_WSTR,2)DAY(GETDATE()),2) + "/" + RIGHT("0" +
(DT_WSTR,2)MONTH(DATEADD("d",31,GETDATE())),2) + "/" +
(DT_WSTR,4)YEAR(GETDATE())
```

1. Convert 8.8.80 or 08.8.80 or 08.08.80 to 08/08/1980

```
1.(DT_DBDate)(RIGHT("0"+SUBSTRING([DateColumn],1,
FINDSTRING([DateColumn],".",1)-1),2) + "/" +
RIGHT("0"+SUBSTRING([DateColumn],FINDSTRING([DateColumn],".",1)+1,
FINDSTRING([DateColumn],".",2)-FINDSTRING([DateColumn],".",1)-1),
2) + "/"+"19"+RIGHT([DateColumn],2))
```

2. Pad a string with zeroes on the left(output length-10)

```
1.RIGHT(("0000000000"+"STRING"),10)      or  
1.REPLICATE("0",10 - LEN("String")) + "STRING"
```

3. Convert date of input string column to date time. Validate that if its value is NULL or empty or illegal date it should be converted to Null date.

```
1.ISNULL([Column 0]) || LEN(TRIM((DT_WSTR,10)[Column 0])) == 0 || [Column 0]==="00-00-0000" NULL(DT_DATE) : (DT_DBDATE)((DT_WSTR,10)[Column 0])
```

4. Create file name with the current time stamp in the format FileName\_YY-MM-DD-HHMMSS.txt

```
1."FileName_" + SUBSTRING((DT_WSTR,30)GETDATE(),1,10) +
SUBSTRING((DT_WSTR,30)GETDATE(),12,2)    SUBSTRING((DT_WSTR,30)GETDATE(),
15,2) + SUBSTRING((DT_WSTR,30)GETDATE(),18,2) + ".txt"
```

5. Get the difference of the date in a column w.r.t. today's date

```
1.DATEDIFF("d", (DT_DATE)(SUBSTRING([Column 0],10,2) + "-" +
SUBSTRING([Column 0],12,2) + "-" + SUBSTRING([Column 0],6,4)),GETDATE())
```

6. If length of a column or variable is zero set its value to "asd" else set it to the variable value.

```
1.LEN([Column 0]) == 0 ? "asd" : [Column 0]
```

FindString function:

FINDSTRING("111<sub>abcda</sub>", "aa", 1) this will give 0

FINDSTRING("111<sub>abcda</sub>", ".a", 1) this will give 4

FINDSTRING("111<sub>abcda</sub>", "a", 2) this will give 8

7. Check if a string contains a value do Step1 else Step2

1. FINDSTRING("111abcda", "aa", 1) > 0 ? Step1 : Step2

8. Set File name in the following format: FileName\_MMMDD\_YYYY.txt (FileName\_Nov23\_2009.txt)

```
1. "FileName_" + (MONTH(GETDATE()) == 1 ? "Jan" : MONTH(GETDATE()) == 2 ? "Feb" :
MONTH(GETDATE()) == 3 ? "Mar" : MONTH(GETDATE()) == 4 ? "Apr" : MONTH(GETDATE()) ==
5 ? "May" : MONTH(GETDATE()) == 6 ? "Jun" : MONTH(GETDATE()) == 7 ? "Jul" :
MONTH(GETDATE()) == 8 ? "Aug" : MONTH(GETDATE()) == 9 ? "Sep" : MONTH(GETDATE()) ==
10 ? "Oct" : MONTH(GETDATE()) == 11 ? "Nov" : MONTH(GETDATE()) == 12 ? "Dec" :
"InvalidMonth") + (DT_WSTR,3)DAY(GETDATE()) + "_" + (DT_WSTR,5)YEAR(GETDATE()) +
".txt"
```

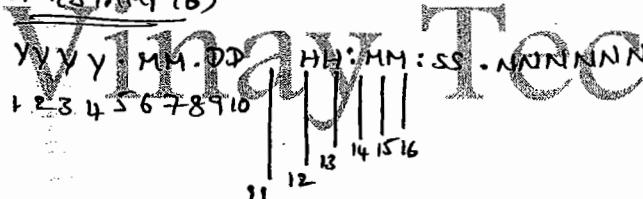
1. What is Time stamp, tell me the need.

i) It shows date & time value.

ii) Max. DWH tables use this for simple and easier operations and time calculations.

iii) There are various timestamps available, industry recommended is 6 timestamp.

TIMESTAMP(6)



2015.01.07 08:20:18.667799

2. Diff b/w DT\_STR and DT\_WSTR

DT\_STR

1. Null terminated ASCII

2. STR → String

Syn:

DT\_STR, length, codepage

Eg: "2012-09-28"

DT\_STR, 10, 1252

When we work with flatfiles,

Databases we get these data types.

DT\_WSTR

1. Null terminated Unicode

WSTR → Wide String

DT\_WSTR, length.

DT\_WSTR, 10

Excel and XML app's we find these

## TRANSFORMATIONS

Transformations are key components within the Data Flow that allow changes to the data within the data pipeline. You can use transformations to split, divert, and remerge data in the data pipeline. Data can also be validated, cleansed, and rejected using specific rules.

**Aggregate:** Aggregates data from transformation or source.

**Audit:** Exposes auditing information from the package to the data pipe, such as when the package was run and by whom.

**CDC Splitter:** After data has been read out of a table with CDC enabled, this transform sends data that should be inserted, updated, and deleted down different paths.

**Character Map:** Makes common string data changes for you, such as changing data from lowercase to uppercase.

**Conditional Split:** Splits the data based on certain conditions being met. For example, this transformation could be instructed to send data down a different path if the State column is equal to Florida.

**Copy Column:** Adds a copy of a column to the transformation output. You can later transform the copy, keeping the original for auditing purposes

**Data Conversion:** Converts a column's data type to another data type.

**Data Mining Query:** Performs a data-mining query against Analysis Services.

**Derived Column:** Creates a new derived column calculated from an expression.

**DQS Cleansing:** Performs advanced data cleansing using the Data Quality Services-engine.

**Export Column:** Exports a column from the Data Flow to the file system. For example, you can use this transformation to write a column that contains an image to a file.

**Fuzzy Grouping:** Performs data cleansing by finding rows that are likely duplicates.

**Fuzzy Lookup:** Matches and standardizes data based on fuzzy logic. For example, this can transform the name Jon to John.

**Import Column:** Reads data from a file and adds it to a Data Flow.

**Lookup:** Performs a lookup on data to be used later in a transformation. For example, you can use this transformation to look up a city based on the zip code.

**Merge:** Merges two sorted data sets into a single data set in a Data Flow.

**Merge Join:** Merges two data sets into a single data set using a join function.

**Multicast:** Sends a copy of the data to an additional path in the workflow.

**OLE DB Command:** Executes an OLE DB command for each row in the Data Flow.

**Percentage Sampling:** Captures a sampling of the data from the Data Flow by using a percentage of the Data Flow's total rows.

**Pivot:** Pivots the data on a column into a more non relational form. Pivoting a table means that you can slice the data in multiple ways, much like in OLAP and Excel.

**Row Count:** Stores the row count from the Data Flow into a variable.

**Row Sampling:** Captures a sampling of the data from the Data Flow by using a row count of the Data Flow's total rows.

**Script Component:** Uses a script to transform the data. For example, you can use this to apply specialized business logic to your Data Flow.

**Slowly Changing Dimension:** Coordinates the conditional insert or update of data in a slowly changing dimension.

**Sort:** Sorts the data in the Data Flow by a given column.

**Term Extraction:** Looks up a noun or adjective in text data.

**Term Lookup:** Looks up terms extracted from text and references the value from a reference table.

**Union All:** Merges multiple data sets into a single data set.

**Unpivot:** Unpivots the data from a non-normalized format to a relational format.

- 2012 onwards available.

## MULTIFLATFILE CONNECTION MANAGER

The Multiple Flat File Connection Manager refers to The list of files for copying or moving or it may hold a series of sql scripts to execute, similar to the file connection manager. The Multiple Flat File Connection Manager gives you the same view as a Flat File Connection Manager, but it enables you to point to multiple files. In either case, you can point to a list of files by placing a vertical bar (|) between each filename: **C:\Projects\011305c.dat|C:\Projects\053105c.dat**.

In the Data Flow, the Multiple Flat File Connection Manager reacts by combining the total number of records from all the files that you have pointed to, appearing like a single merged file. Using this option will initiate the Data Flow process only once for the files whereas the Foreach Loop container will initiate the process once per file being processed. In either case, the metadata from the file must match in order to use them in the Data Flow. Most developers lean toward using Foreach Loop Containers because it's easier to make them dynamic. With these Multiple File or Multiple Flat File Connection Managers, you have to parse your file list and add the vertical bar between them. If you use Foreach Loop Containers, that is taken care of for you.

## SYNCHRONOUS Versus ASYNCHRONOUS TRANSFORMATIONS

Transformations are divided into two main categories: a) Synchronous b) Asynchronous

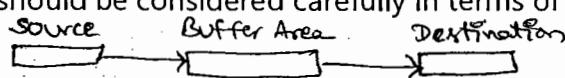
- Synchronous transformations are components such as the Derived Column and Data Conversion Transformations, where rows flow into memory buffers in the transformation and add the same buffers come out. No rows are held, and typically these transformations perform very quickly with minimal impact to your data flow.
- Asynchronous transformations can cause a block in your Data Flow and slow down your runtime.

There are two types of **asynchronous** transformations: Partially blocking and fully blocking.

Partially blocking transformations, such as the Union All, create new memory buffers for the output of the transformation.

Fully blocking transformations, such as the Sort and Aggregate Transformations, do the same thing but cause a full block of the data.

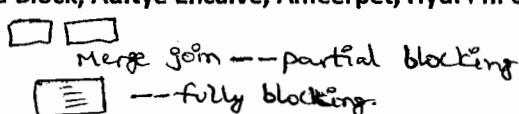
In order to sort the data, SSIS must first see every single row of the data. If you have a 100MB file, then you may require 200MB of RAM in order to process the Data Flow because of a fully blocking transformation. These fully blocking transformations represent the single largest slowdown in SSIS and should be considered carefully in terms of any architecture decisions you must make.



i/p buffers = o/p buffers => Synchronous

i/p buffers <> o/p buffers => Asynchronous

# 505, Annapurna Block, Aditya Enclave, Ameerpet, Hyd. Ph: 04066638869/9573168449



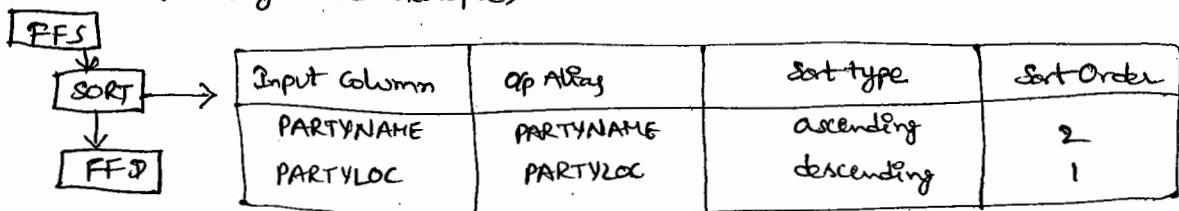
05-01

**SORT**

Fully asynchronous transform (fully blocking)

1. This Transformation is a fully blocking asynchronous transformation that enables you to sort data based on any column in the path.
2. You can optionally check the \_\_\_\_\_ option to "Remove rows that have duplicate sort values." This is a great way to do rudimentary de-duplication of your data. If a second value comes in that matches your same sort key, it is ignored and the row is dropped.
3. Supports multi-column sorting

Ex & Pic: Display Locations in descending order, within the location names in ascending order (Multi-column sorting order example).

**DATA CONVERSION WAYS AND TRANSFORMATION**

The Data Conversion Transformation performs a similar function to the CONVERT or CAST functions in T-SQL.

**Types:** a) By using data conversion transformation.

Refer to the aggregate transformation ex  
b) By using Typecast operators in expressions

Syn: [Typecast Operator] <exp>

Ex: [DT\_14] 12345

c) By directly using at source level

This eliminates intermediate conversions multiple times.

Goto source → Show advanced editor → I/p O/p properties → Flat file source

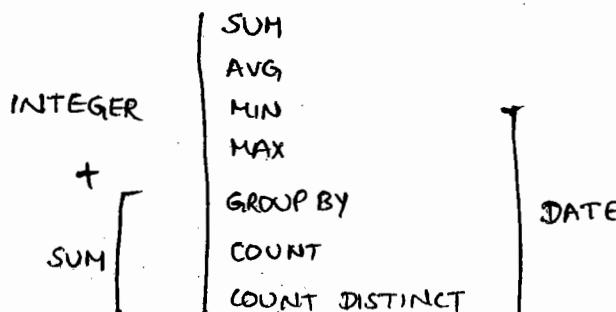
O/p → Column name → Data Type: DT\_

05/11

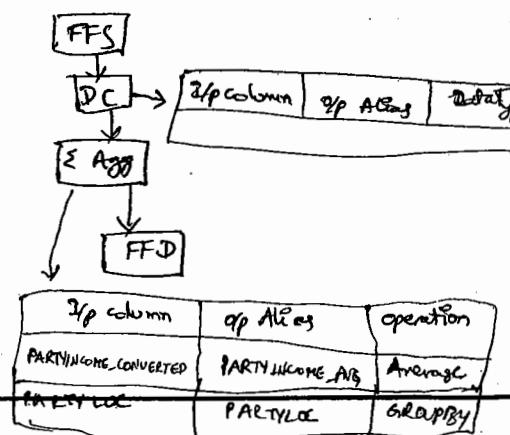
**AGGREGATE TRANSFORMATION**

The fully blocking asynchronous Aggregate Transformation allows you to aggregate data from the Data Flow to apply certain T-SQL functions that are done in a GROUP BY statement, such as sum, avg, min, max, count distinct, group by, count.

Ex:



Ex: Display location wise sum and avg of Income



## DERIVED COLUMN

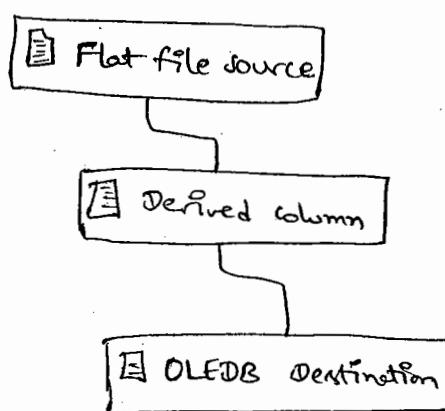
The Derived Column Transformation creates a new column i.e., calculated or derived from the other column(s) or set of columns. It is one of the most important transformations in your Data Flow arsenal.

Most frequently used row-wise synchronous transform:

Ex: Implement the below business operations on the source data.

- Concatenate Name and location with '-' and load as address.
- Create Income increment with 10% of Income value.
- Add Business\_date column as current\_date
- Load Income as 9999 if Income is NULL or empty

**Ex & Pic:**



Derived column Name	Derived column	Expression.
Address	Creates new column>	PARTYNAME + "-" + PARTYLOC
PINC_INC		(DT_I4)PARTYINCOME + (DT_I4)
BUSINESS_DATE		PARTYINCOME * 10/10
COMPANY_CODE		GETDATE()
TIME_INFO		21000
PATH		SUBSTRING(DT_WSTR, 30)
USER PACKAGE		GETDATE(1,12,8)
PARTYINCOME	Replace "PARTYINCOME"	INPUTPARTY_SRC.TX Q[System::UserName]+ "-" + @ [System::PackageName] ISNULL(PARTYINCOME) ? "9999" PARTYINCOME

BDW TABLENAME	BDW COLUMN NAME	FILENAME	COLUMNNAME	Mapping REMARKS	Transformation Rule	Sort Order
PARTY	Party Id	DATA_VARTEXT	Organization,Number	See mapping rule	Organization,Number concatenated with '_'	2
PARTY	Partyname	DATA_VARTEXT	Partyname	See mapping rule	Concatination of firstname,middlename and last name.	
PARTY	Email	DATA_VARTEXT		Hard Coded value	<a href="mailto:msbivinay@yahoo.co.in">msbivinay@yahoo.co.in</a>	
PARTY	Partycode	DATA_VARTEXT	Partycode	See mapping rule	<u>DIRECT MOVE</u>	1
PARTY	Partyincome	DATA_VARTEXT	partyincome	See mapping rule	<u>DIRECT MOVE</u>	
PARTY	Partyincome_Inc	DATA_VARTEXT	partyincome_inc	PI value incrementation	pi*12/100	
PARTY	Source_System_ID			See mapping rule	Set the value as '41000'	

OSI"

**CONDITIONAL SPLIT**

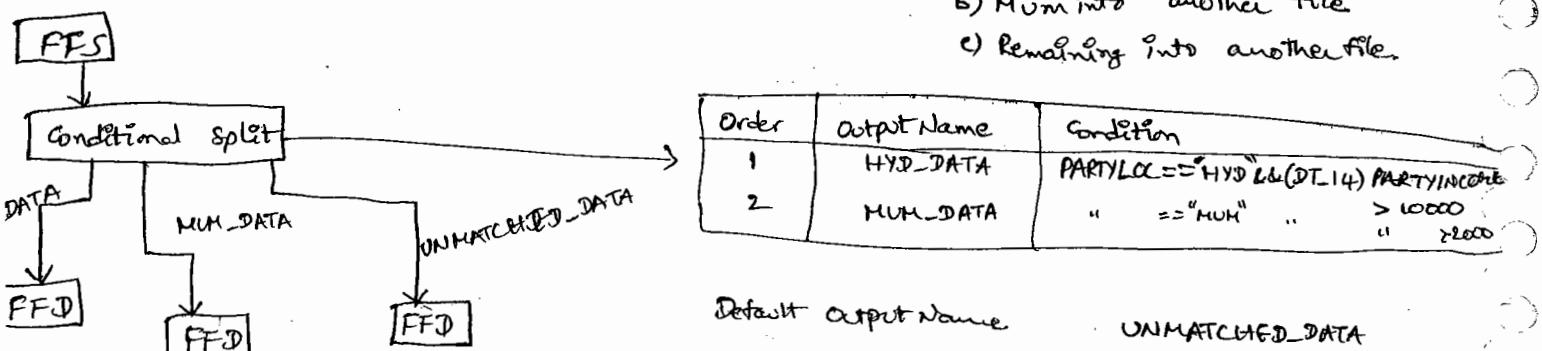
Splits the database on the condition.  
Generally it gives 2 types of op.

- a) Condition matched op      b) Condition unmatched op.

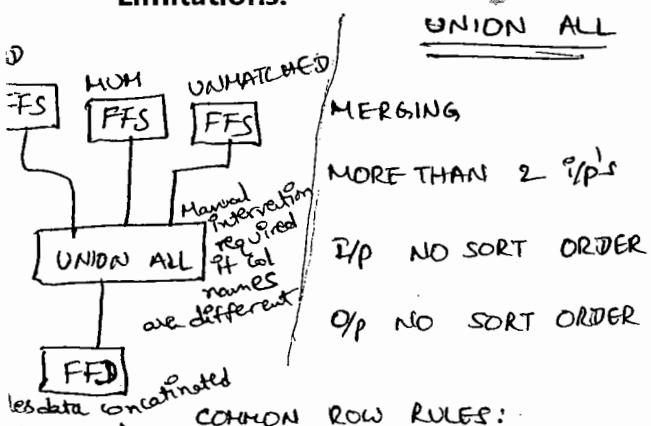
Eg: Split the party source file data into 3 locations. a) Hyd into 1 file

b) Mum into another file

c) Remaining into another file.

**UNION ALL**

The Union All Transformation works much the same way as the Merge Transformation, but it does not require sorted data. It takes the outputs from multiple source transformations and combines them into a single result set.

**Limitations:**

emp  
=====

eid(int), enm(varchar(30))

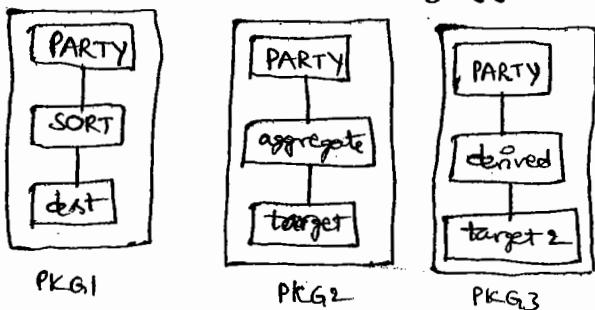
emp1  
=====

empid(int), pname(varchar(30))

## MULTICAST

The Multicast Transformation, as the name implies, can send a single data q/p to multiple o/p path easily. You may want to use this transformation to send a path to multiple destinations sliced in different ways. To configure this transformation, simply connect it to your input, and then drag the output path from the Multicast Transformation onto your next destination or transformation. After you connect the Multicast Transformation to your first destination or transformation, you can keep connecting it to other transformations or destinations. There is nothing to configure in the Multicast Transformation Editor other than the names of the outputs.

### Drawbacks with Existing Approach

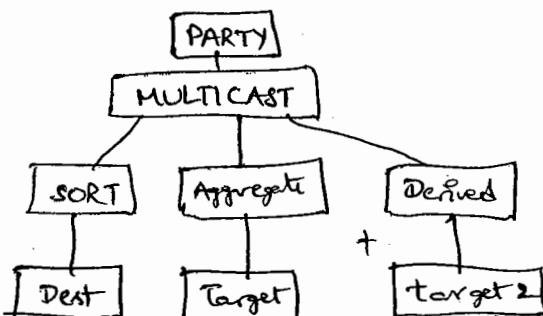


#### Drawbacks

1. 3 times hit on source
2. Pkg resources more
3. More buffer occupied
4. Burden on the SSIS server
5. Traffic high.

### Advantage with Multicast Approach

Single source passing Transformation.



#### Advantages:

1. 1 time hit on the sources
2. Less pkg resources
3. Less buffer
4. Less burden on the SSIS server
5. Traffic is less

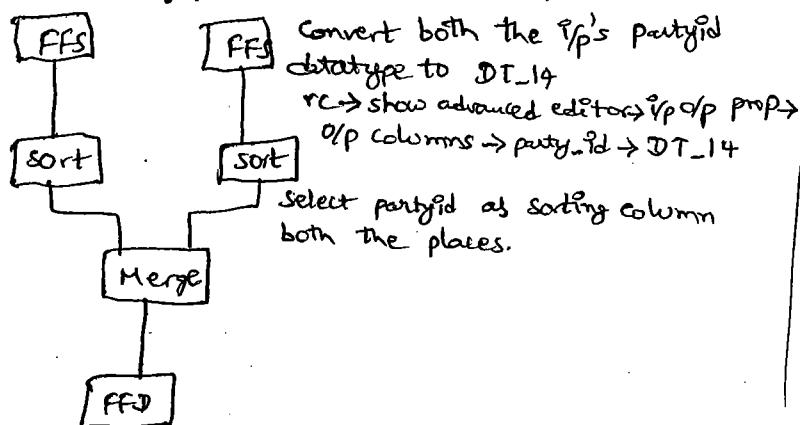
## MERGE

The Merge Transformation can merge data from two paths into a single q/p. This transformation is useful when you wish to break out your Data Flow into a path that handles certain errors and then merge it back into the main Data Flow downstream after the errors have been handled. It's also useful if you wish to merge data from two Data Sources.

This transformation is similar to the Union All Transformation, but the Merge Transformation has some restrictions that may cause you to lean toward using Union All:

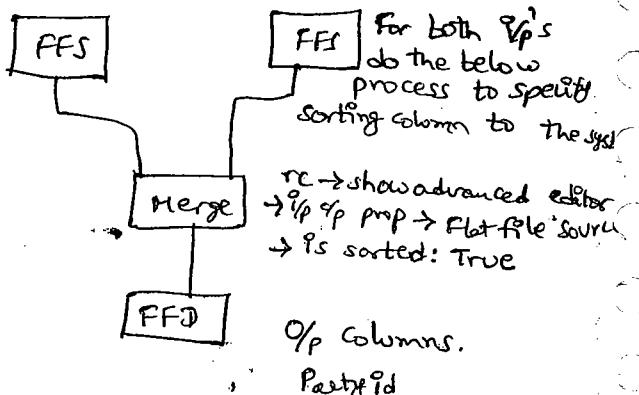
**Two Ways :**

1<sup>st</sup> way (If sources are not sorted)



2<sup>nd</sup> way (If sources are already sorted)

We need to tell to the system the file is sorted for so and so column.



Sort key position: 1

DataType: DT\_14.

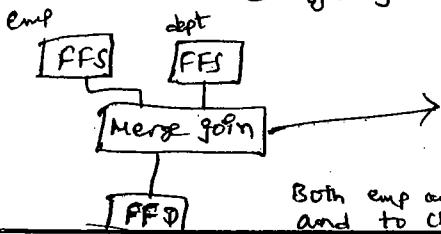
## MERGE JOIN

This transformation will merge the output of two inputs and perform an INNER or OUTER join on the data. An example of when this would be useful is if you have a front-end web system in one data stream that has a review of a product in it, and you have an inventory product system in another data stream with the product data. You could merge join the two data inputs and output the review and product information into a single path.

The Merge Join Transformation can join large volumes of data without much memory impact; however, it has certain requirements, such as sorted input columns, that may be difficult to meet. Remember to use the source query to sort the input data, and avoid the Sort Transformation when possible, because of performance issues.

**Practicals:** Merge + Join means while merging the data it performs join operation. The below joins will supported  
 a) inner join  
 b) left outer join  
 c) full outer join.

NOTE: There is no outer join since if you swap the %p's and perform left join then it could be right join.



Left	NAME	order	Join key
emp	EID	0	<input type="checkbox"/>
dept	ENAME	0	<input type="checkbox"/>
	DID	1	<input checked="" type="checkbox"/>

Right	Name
	DID
	DNAME

Both emp and dept did column to use for sort key position  
 and to change datatype to DT\_14.

**JOINS**

EID	ENM	DID
1	VIN	10
2	MAD	20

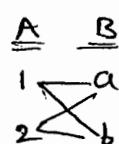
EMP

DID	DNM
10	IT
30	HR

DEPT

E.EID E.ENM E.DID D.DID D.DNM

Cross Join



1	VIN	10	10	IT
1	VIN	10	30	HR
2	MAD	20	10	IT
2	MAD	20	30	HR

Inner Join



1	VIN	10	10	IT
---	-----	----	----	----

Left Join



1	VIN	10	10	IT
2	MAD	20	?	?

Right Join



1	VIN	10	10	IT
?	?	?	30	HR

Full Join



1	VIN	10	10	IT
2	MAD	20	?	?
?	?	?	30	HR

Joins: Multiple cols from multiple tables

Types:

No condition join (Cross join)

Matched join (Inner join) equi  
non equi  
self

Matched and Unmatched

Join

(Outer Join)

Left Outer

Right Outer

Full Outer

Sel e.eid,e.enm,e.did,d.did,d.dnm  
from emp e  
cross join dept d  
inner join dept d on e.did=d.did  
left join dept d on e.did=d.did  
right join dept d on e.did=d.did  
full join dept d on e.did=d.did

Diff b/w Merge join and LOOKUP?

### MSBI 2012 (SQL Server Integration Services)

48

What is the structure of LOOKUP unmatched data? Similar to source structure.

### LOOKUP

**Operation:** Perform an exact match and get required columns.

**Real time Usage:** 1. To verify the row existence.

2. To identify duplicates.

3. To implement slowly changing dimension load.

4. Change data capturing loading.

**No match in lookup table** There are 4 options.

a) Ignore failure

b) Fail component

c) Re-direct rows to no match output

d) Re-direct rows to error output

If multiple matches found always it returns first match.

(A)	Merge join	LOOKUP
	<ul style="list-style-type: none"><li>I/p's sorting order</li><li>Exact match through innerjoin</li><li>No cache</li><li>Can operate on file</li><li>Supports outer join</li></ul>	<ul style="list-style-type: none"><li>No restriction</li><li>Defined for exact match</li><li>Supports cache</li><li>Lookup can not be file</li><li>N/A</li></ul>

**Lookup Objects** Can be a table (or) a cache

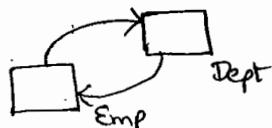
**Types of Caches:** To reduce the burden on the server (no cache). There are

2 Caches introduced

a) full cache      b) partial cache

# Vinay Tech House

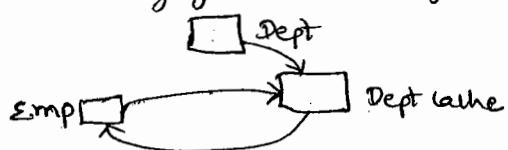
No Cache: Always hits the destination table, if the data changing frequently this is useful.



Full Cache: Always requests hits the cache only.

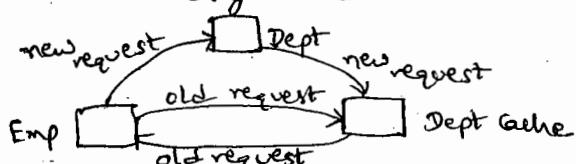
When you run the pkg first system creates cache and it uses for its lookup operation.

If the data is not changing such a long time then it is recommended.



Partial Cache: Initially cache empty for every request source hits the table to get result into cache and destination, for old request hits the cache to fetch the data.

Advantage if new records adding more and more on the destination.



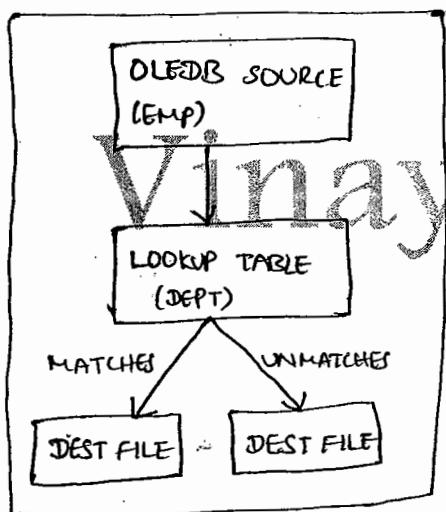
### Concept

The Lookup Transformation performs what equates to an INNER JOIN on the Data Flow and a second data set. The second data set can be an OLE DB table or a cached file, which is loaded in the Cache Transformation. After you perform the lookup, you can retrieve additional columns from the second column. If no match is found, an error occurs by default. You can later choose, using the Configure Error Output button, to ignore the failure (setting any additional columns retrieved from the reference table to NULL) or redirect the rows down the second non matched green path.

### Cache

The Cache Transformation enables you to load a cache file on disk in the Data Flow. This cache file is later used for fast lookups in a Lookup Transformation. The Cache Transformation can be used to populate a cache file in the Data Flow as a transformation, and then be immediately used, or it can be used as a destination and then used by another package or Data Flow in the same package. The cache file that's created enables you to perform lookups against large data sets from a raw file. It also enables you to share the same lookup cache across many Data Flows or packages.

### PRACTICALS ( NO / FULL / PARTIAL / SHARED )



Common settings for LOOKUP all operations.  
Full cache implementation.

a) RC →

General  
Select @Full cache  
LOOKUP unmatched records  
Required to no match destination  
Connection:  
OLEDB destination  
Connection:  
Specify database name  
Lookup table: Dept  
Columns  
Specify matching column and retrieval column  
OK

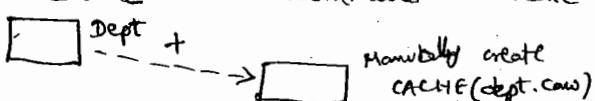
Partial cache implementation

Similar to above process with small replacement  
that is full cache to partial cache.

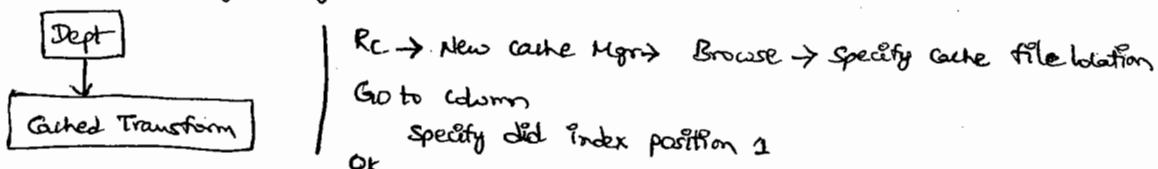
NO cache implementation

Similar to above with No cache option selection.

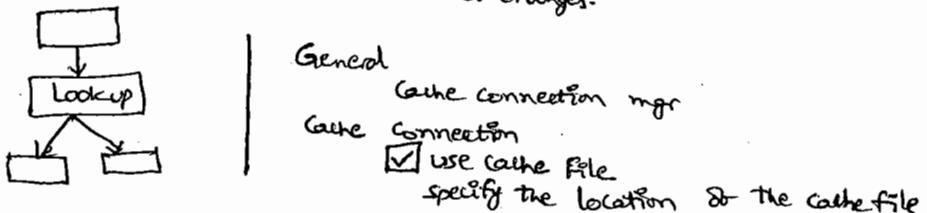
Shared cache: If a table is used by multiple other tables LOOKUP and the table data is not changing for some time it is recommended to create shared cache, and use across all pkg's.



1. Create a Cache by using the Cache Transform like below



2. Same pkg like earlier with minor changes.



Note: Cache files are extended with .caaw.

## FUZZY LOOKUP

# Vinay Tech House

Using the Fuzzy Lookup Transformation requires an input stream of at least one field that is a string. Internally, the transformation has to be configured to connect to a reference table that will be used for comparison. The output to this transformation will be a set of columns containing the following:

**Input and Pass-Through Field Names and Values:** This column contains the name and value of the text input provided to the Fuzzy Lookup Transformation or passed through during the lookup.

**Reference Field Name and Value:** This column contains the name and value(s) of the matched results from the reference table.

**Similarity:** This column contains a number between 0 and 1 representing similarity to the matched row and column. Similarity is a threshold that you set when configuring the Fuzzy Lookup Task. The closer this number is to 1, the closer the two text fields must match.

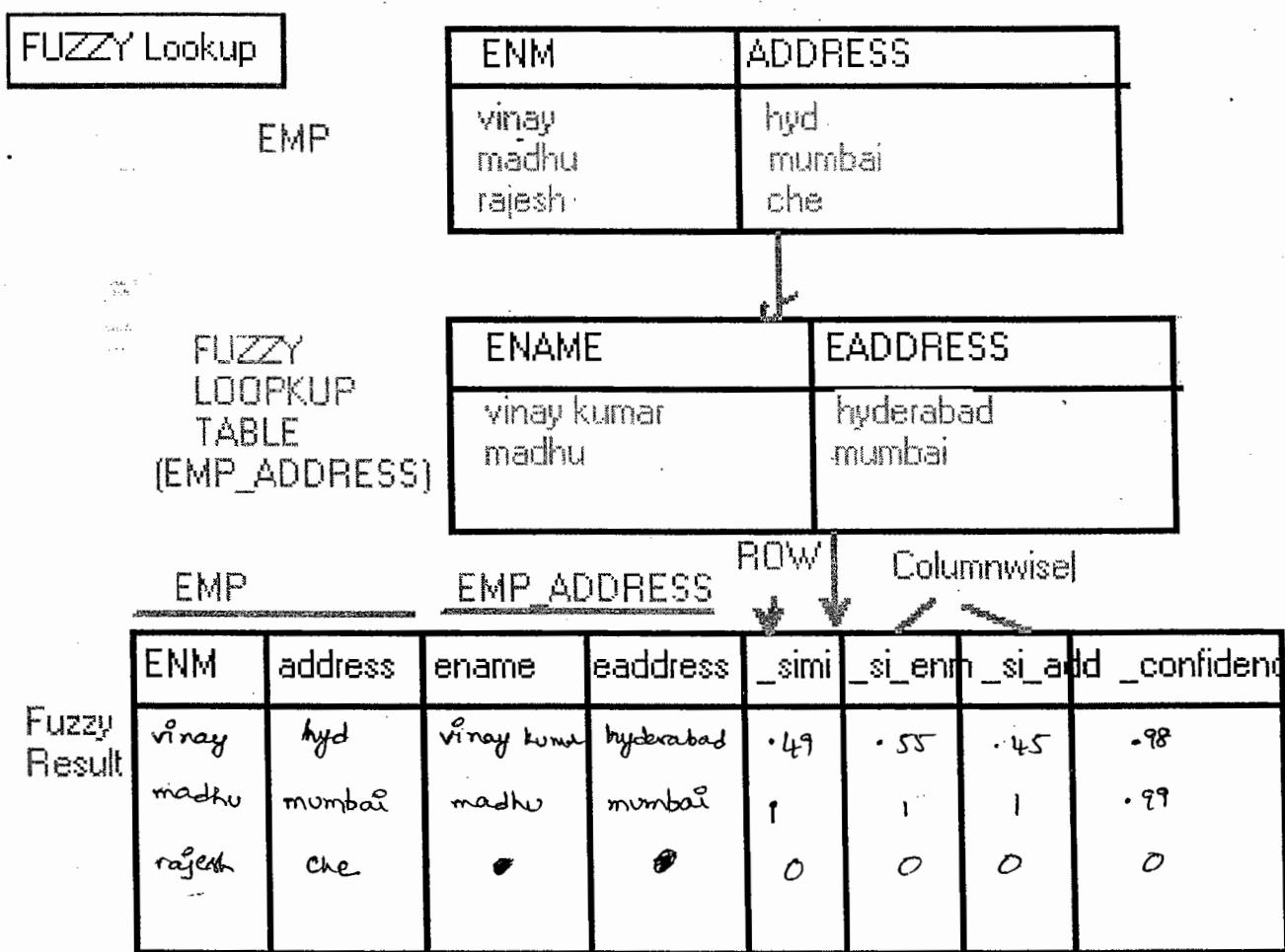
**Confidence:** This column contains a number between 0 and 1 representing confidence of the match relative to the set of matched results. Confidence is different from similarity, because it is not calculated by examining just one word against another but rather by comparing the chosen word match against all the other possible matches. For example, the value of Knight Brian may have a low similarity threshold but a high confidence that it matches to Brian Knight. Confidence gets better the more accurately your reference data represents your subject domain, and it can change based on the sample of the data coming into the ETL process.

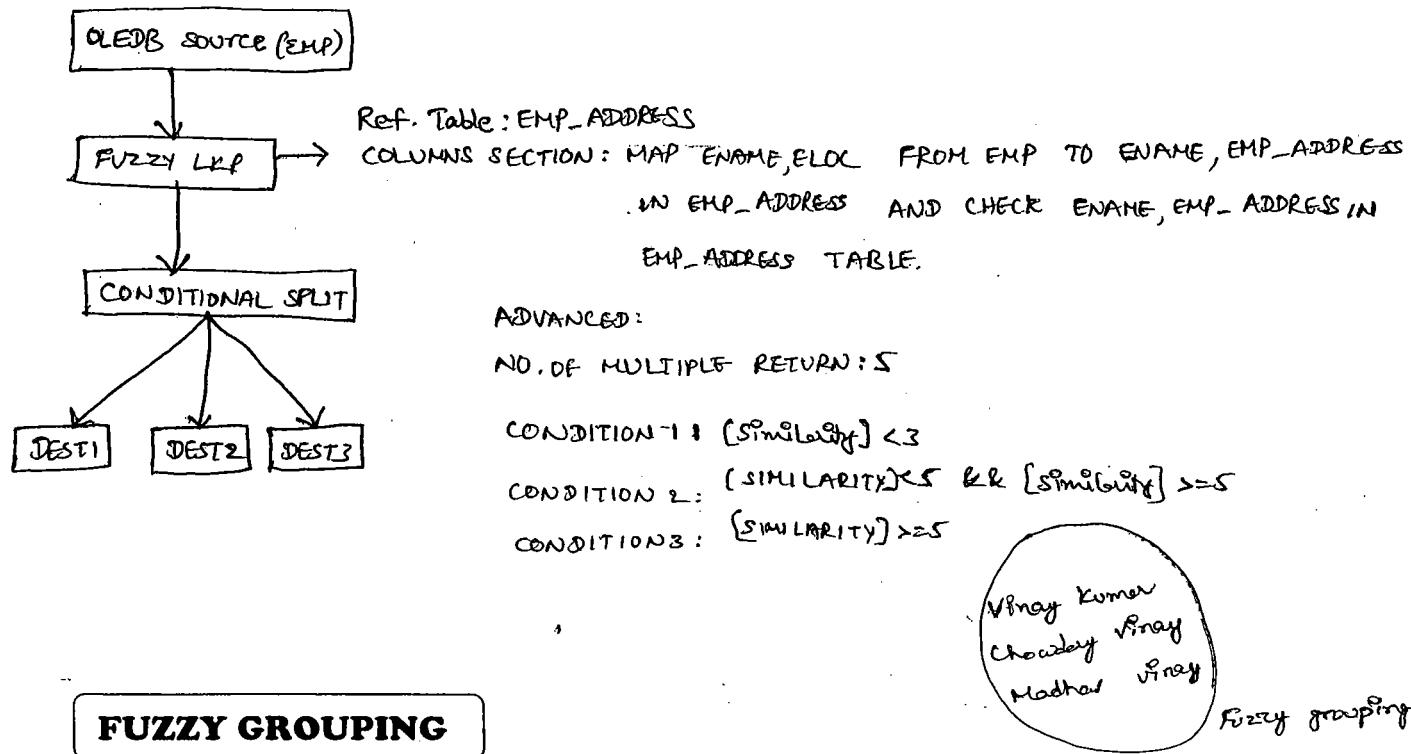
*There are a few additional settings in this tab that are of interest.*

The default option to set is the “Generate new index” option. By setting this, a table will be created on the reference table’s Connection Manager each time the transformation is run, and that table will be populated with loads of data as mentioned earlier in this section. The creation and loading of the table can be an expensive process. This table is removed after the transformation is complete. Alternatively, you can select the “Store new index” option, which will instantiate the table and not drop it. You can then

reuse that table from other Data Flows or other Data Flows from other packages and over multiple days. As you can imagine, by doing this your index table becomes stale soon after its creation. There are stored procedures you can run to refresh it in SQL, or you can click the “Maintain stored index” checkbox to create a trigger on the underlying reference table to automatically maintain the index table. This is available only with SQL Server reference tables, and it may slow down your insert, update, and delete statements to that table.

**Advanced:** This tab contains the settings that control the fuzzy logic algorithms. You can set the maximum number of matches to output per incoming row. The default is set to 1, which means pull the best record out of the reference table if it meets the similarity threshold. Incrementing this setting higher than this may generate more results that you'll have to sift through, but it may be required if there are too many closely matching strings in your domain data. A slider controls the Similarity threshold. It is recommended that you start this setting at .71 when experimenting and move up or down as you review the results. This setting is normally determined based on a businessperson's review of the data, not the developer's review. If a row cannot be found that's similar enough, the columns that you checked in the Columns tab will be set to NULL. The token delimiters can also be set if, for example, you don't want the comparison process to break incoming strings up by a period (.) or spaces.





## FUZZY GROUPING

The Fuzzy Grouping Transformation can look through a list of similar text and group the results using the same logic as fuzzy lookup. You can use these groupings in a transformation table to clean up source and destination data or to crunch fact tables into more meaningful results without altering the underlying data. The Fuzzy Group Transformation also expects an input stream of text, and it requires a connection to an OLEDB Data Source because it creates in that source a set of structures to use during analysis of the input stream.

## PIVOT TRANSFORM

(denormalize to normalize)

A pivot table is a result of cross tabulated columns generated by summarising data from a row format. Typically, a Pivot Transformation is generated using the following input columns:

**Pivot Key:** A pivot column is the element of input data to "pivot." The word "pivot" is another way of saying "to create a column for each unique instance of." However, this data must be under control. Think about creating columns in a table. You wouldn't create 1,000 uniquely named columns in a table, so for best results when choosing a data element to pivot, pick an element that can be run through a GROUP BY statement that will generate 15 or fewer columns. If you are dealing with dates, use something like a DATENAME function to convert to the month or day of the year.

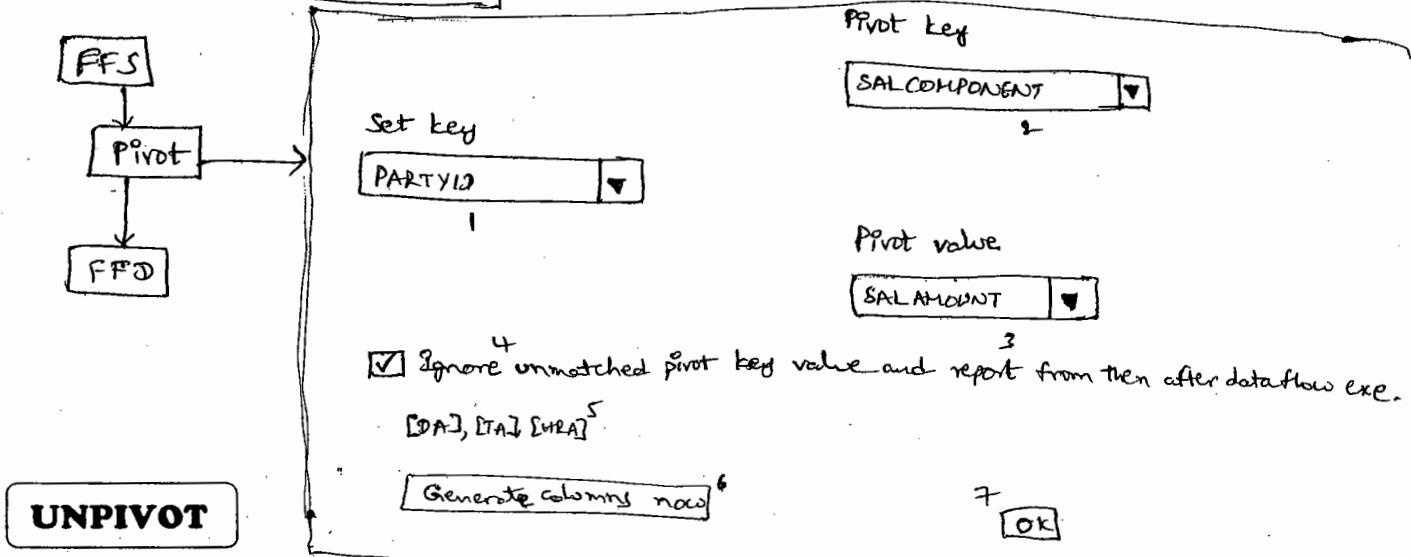
**Set Key:** Set key creates one column and places all the unique values for all rows into this column. Just like any GROUP BY statement, some of the data is needed to define the group (row), whereas other data is just along for the ride.

**Pivot Value:** These columns are aggregations for data that provide the results in the matrix between the row columns and the pivot columns.

PARTY

PARTY ID	SALCOMPONENT	SALAMOUNT
1	HRA	10000
1	DA	10000
1	TA	10000
2	HRA	20000
2	DA	20000
2	TA	20000

PARTYID	HRA	DA	TA
1	10000	10000	10000
2	20000	20000	20000



As you know, mainframe screens rarely conform to any normalized form. For example; a screen may show a Bill To Customer, a Ship To Customer, and a Dedicated To Customer field. Typically, the Data Source would store these three fields as three columns in a file (such as a virtual storage access system, or VSAM). Therefore, when you receive an extract from the mainframe you may have three columns, as shown in Figure 5-37. Your goal is to load this file into a Customer table in SQL Server. You want a row for each customer in each column, for a total of six rows in the CustomerName and OrderID columns in Figure 5-38.

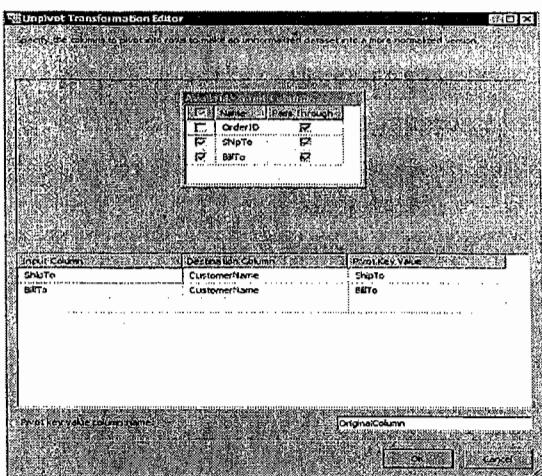
Preview Query Results

Query Result (Up to the first 200 rows)

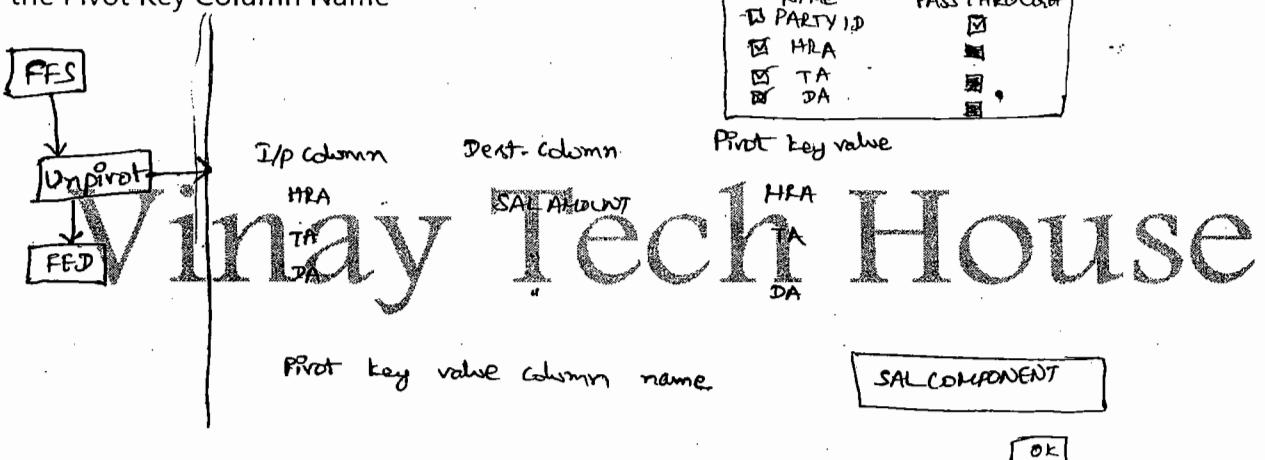
OriginalColumn	CustomerName	OrderID
BillTo	Jennifer Knight	1
ShipTo	Brian Knight	1
BillTo	Bailey McDonald	2
ShipTo	Sherri McDonald	2
BillTo	Manfred Kline	3
ShipTo	Kylie McDonald	3

UNPIVOT → Converting column values to row values  
(Normalize to denormalize)

In this example, you want to unpivot the BillTo and ShipTo columns, and the OrderID column will just be passed through for each row. To do this, check each column you wish to unpivot, as shown in Figure 5-40, and check Pass Through for the OrderID column.



As you check each column that you wish to unpivot on, the column will be added to the grid below (shown in Figure 5-40). You'll then need to type **CustomerName** for the Destination Column property for each row in the grid. This will write the data from each of the two columns into a single column called CustomerName. Optionally, you can also type **Original Column** for the Pivot Key Column Name.



**AUDIT**

To Add auditing information to your dataflow.

The Audit Transformation allows you to track who inserted data into a table and when. Because of acts such as HIPPA and Sarbanes-Oxley (SOX) governing audits, you often must be able to track who inserted data into a table and when. This transformation helps you with that function. The task is easy to configure. Following are some of the available options:

**Execution instance GUID:** GUID that identifies the execution instance of the package

**Package ID:** Unique ID for the package

**Package name:** Name of the package

**Version ID:** Version GUID of the package

**Execution start time:** Time the package began

**Machine name:** Machine on which the package ran

**User name:** User who started the package

**Task name:** Data Flow Task name that holds the Audit Task

**Task ID:** Unique identifier for the Data Flow Task that holds the Audit Task

ID	Source	Auditing	DEST
1	Name	<input checked="" type="checkbox"/> pkg	
2	y	<input checked="" type="checkbox"/> username	

without Auditing  
Source → Derived column → Dest  
(derived add system variable, pkg name, username).

**CHARACTER MAP**

Common character transformation in the flow.

The Character Map Transformation performs. This simple transformation can be configured in a single tab. To do so, check the columns you wish to transform. Then, select whether you want this modified column to be added as a new column or whether you want to update the original column. You can give the column a new name under the Output Alias column. Lastly, select the operation you wish to perform on the inputted column. The available operation types are as follows:

**Byte Reversal:** Reverses the order of the bytes. For example, for the data 0x1234 0x9876, the result is 0x4321 0x6789. This uses the same behavior as LCMapString with the LCMAP\_BYTEREV option.

**Full Width:** Converts the half-width character type to full width.

**Half Width:** Converts the full-width character type to half width.

**Hiragana:** Converts the Katakana style of Japanese characters to Hiragana.

**Katakana:** Converts the Hiragana style of Japanese characters to Katakana.

**Linguistic Casing:** Applies the regional linguistic rules for casing.

**Lowercase:** Changes all letters in the input to lowercase.

**Traditional Chinese:** Converts the simplified Chinese characters to traditional Chinese.

**Simplified Chinese:** Converts the traditional Chinese characters to simplified Chinese.

**Uppercase:** Changes all letters in the input to uppercase.

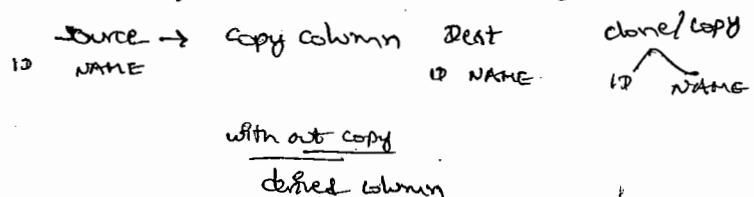
ID	Source	CHARACTER MAP	DEST
1	NAME		
2	Xyz	NAME:UPPER	XYZ

without character map

Source → derived column → Dest  
(upper(name)).

### COPY COLUMN

The Copy Column Transformation is a very simple transformation that copies the value of a column to a clone of itself. This is useful to create a copy of a column before you perform some elaborate transforms. You could then keep the original value as your control subject and the copy as the modified column. To configure this transformation, go to the Copy Column Transformation Editor and check the column you want to clone. Then assign a name to the new column.



### DQS CLEANSING

The Data Quality Services (DQS) Cleansing Transformation performs advanced data cleansing on data flowing through it. With this transformation, you can have your business analyst (BA) create a series of business rules that declare what good data looks like in the Data Quality Client (included in SQL Server). The BA will use a tool called the Data Quality Client to create domains that define data in your company, such as what a Company Name column should always look like. The DQS Cleansing Transformation can then use that business rule.

# Vinay Tech House

### ROW COUNT

The Row Count Transformation provides the capability to count the number of rows in a stream. Each Row Count Transformation is designed for an input stream and will output a row count into a Long (integer) or compatible data type. You can then use this variable to log information into storage, to build e-mail messages, or to conditionally run steps in your packages. Eg: If the number of rows more than 1000 then run script task.

NOTE: To use the value of a row count we need one value to capture.

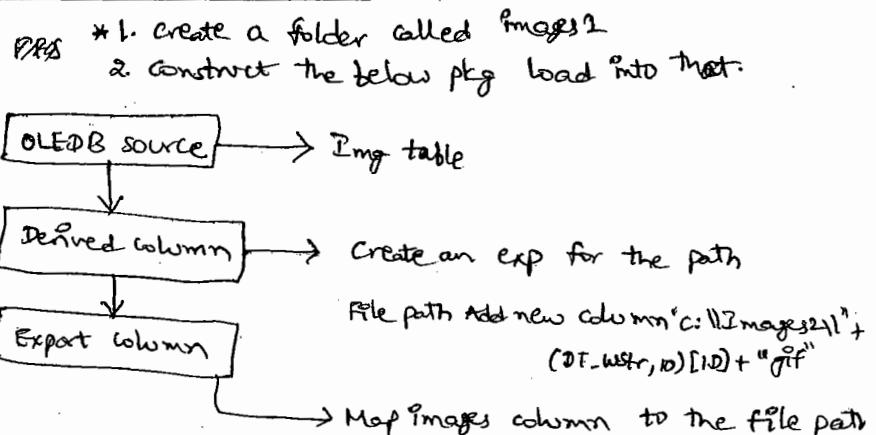
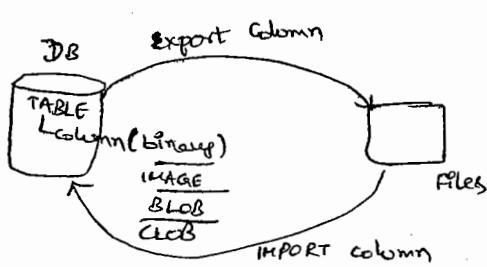
Q1 In which situation you come across Import, Export column transformation?

## MSBI 2012 (SQL Server Integration Services)

57

### EXPORT COLUMN

The Export Column Transformation is a transformation that exports data to a file from the data flow. Unlike other transformations export column does not require a destination to create a file.



Sol: We have a situation like loading files into a table, tracking the files info. into an auditing table.



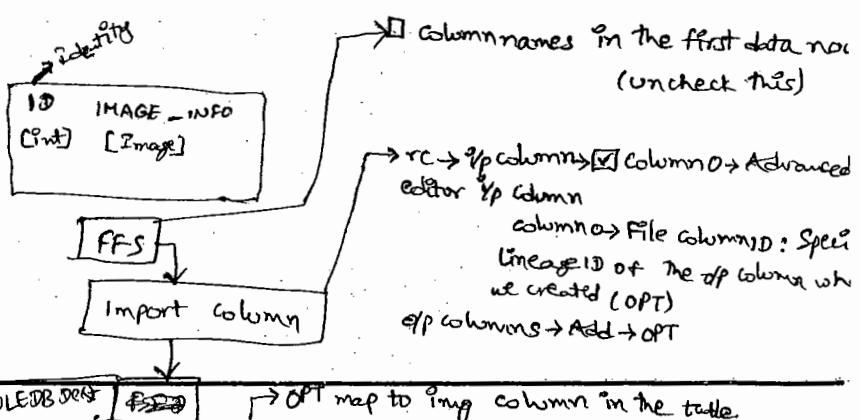
### IMPORT COLUMN

The Import Column Transformation is a partner to the Export Column Transformation. These transformations do the work of translating physical files from system file storage paths into database blob-type fields, and vice versa. The trick to understanding the Import Column Transformation is knowing that its input source requires at least one column that is the fully qualified path to the file you are going to store in the database, and you need a destination column name for the output of the resulting blob and file path string.

1. Take **3** images (preferably gif extension) in a folder.
2. Create a text file like below with images navigation.

```
C:\1\IMAGES\19.gif
C:\1\IMAGES\20.gif
C:\1\IMAGES\21.gif
```

3. Create a table like below.



## OLEDB COMMAND

The OLE DB Command Transformation is a component designed to execute a sql st. for each row in an input stream. This task is analogous to an ADO Command object being created, prepared, and executed for each row of a result set. The input stream provides the data for parameters that can be set into the SQL statement, which is either an in-line statement or a stored procedure call. There are four tabs in the editor:

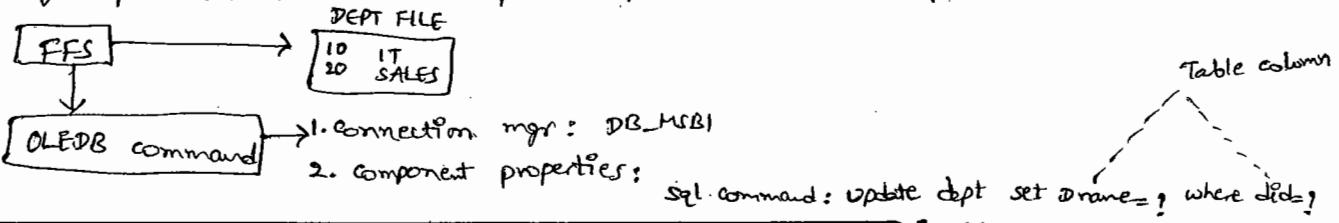
**Connection Manager:** Allows the selection of an OLE DB Connection. This connection is where the SQL statement will be executed. This doesn't have to be the same connection that is used to provide the input stream.

**Component Properties:** Here you can set the SQL Command statement to be executed in the SqlCommand property, and set the amount of time to allow for a timeout in the CommandTimeout property, in seconds. The property works the same way as the ADO Command object. The value for the CommandTimeout of 0 indicates no timeout. You can also name the task and provide a description in this tab.

- **Column Mappings:** This tab displays columns available in the input stream and the destination columns, which will be the parameters available in the SQL command. You can map the columns by clicking a column in the input columns and dragging it onto the matching destination parameter. It is a one-to-one mapping, so if you need to use a value for two parameters, you need use a Derived Column Transformation to duplicate the column in the input stream prior to configuring the columns in this transformation.

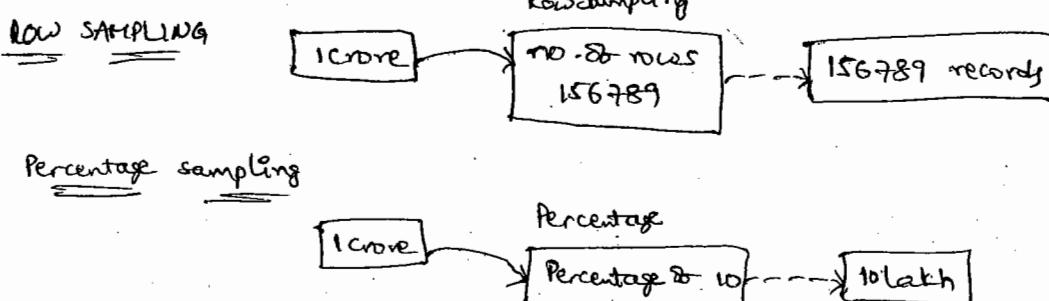
**Input and Output Properties:** Most of the time you'll be able to map your parameters in the Column Mappings tab. However, if the OLE DB provider doesn't provide support for deriving parameter information (parameter refreshing), you have to come here to manually set up your output columns using specific parameter names and DBParamInfoFlags.

Eg: Using Department file data to update dept names in the dept table.



## PERCENTAGE AND ROW SAMPLING

The Percentage Sampling and Row Sampling Transformations enable you to take the data from the source and randomly select a subset of data. The transformation produces two outputs that you can select. One output is the data that was randomly selected, and the other is the data that was not selected.



## TERM EXTRACTION

The Advanced tab, which allows for some configuration of the task, is divided into four categories:

**Term Type:** Settings that control how the input stream should be broken into bits called *tokens*.

The Noun Term Type focuses the transformation on nouns only, Noun Phrases extracts noun phrases, and Noun and Noun Phrases extracts both.

**Score Type:** Choose to analyze words either by frequency or by a weighted frequency.

**Parameters:** Frequency threshold is the minimum number of times a word or phrase must appear in tokens. Maximum length of term is the maximum number of words that should be combined together for evaluation.

**Options:** Check this option to consider case sensitivity or leave it unchecked to disregard.

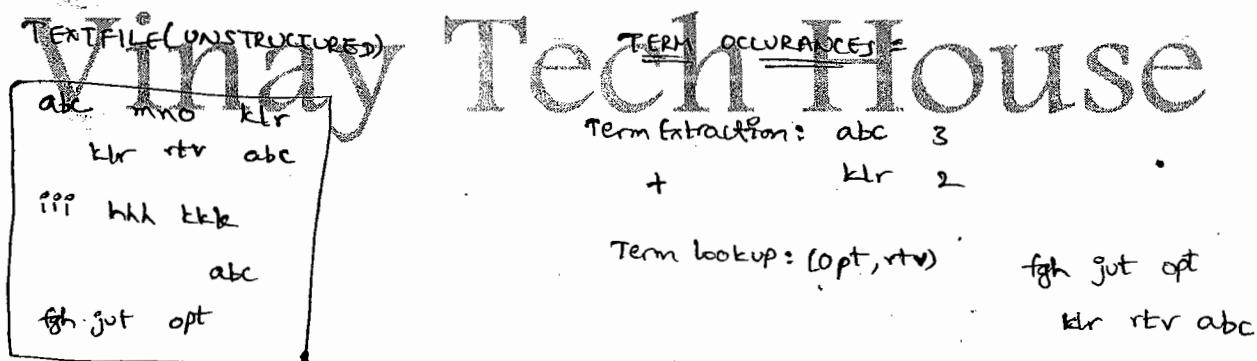
## TERM LOOKUP

Three basic tabs are used to set up this task (in the Term Lookup Transformation Editor):

**Reference Table:** This is where you configure the connection to the reference table. The Term Lookup Task should be used to validate each tokenized term that it finds in the input stream.

**Term Lookup:** After selecting the lookup table, you map the field from the input stream to the reference table for matching.

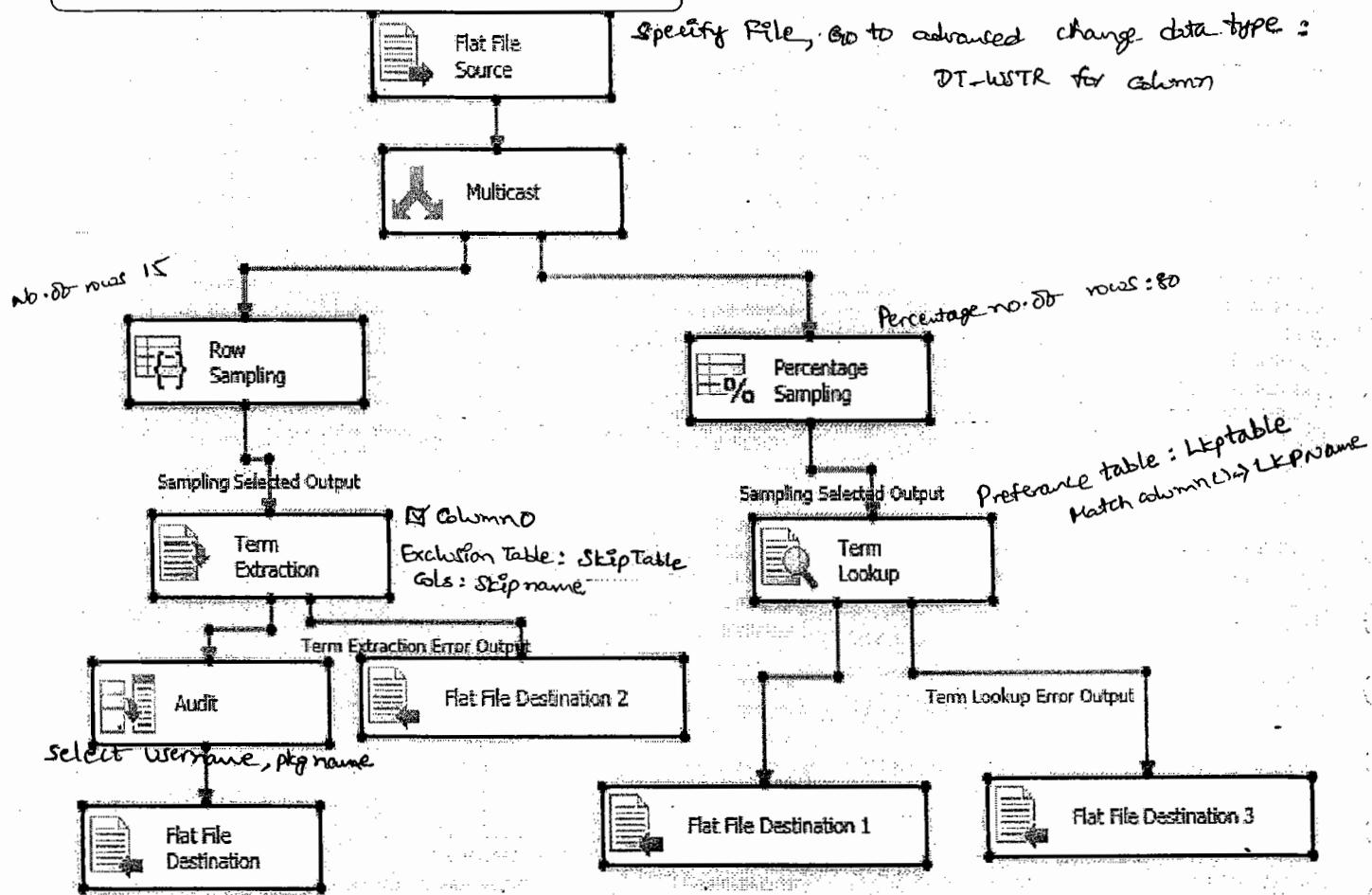
**Advanced:** This tab has one setting to check whether the matching is case sensitive.



## Complex Examples:

### A . Dynamic Flat File Connection Loading

## CONSOLIDATION OF ALL TASKS



1. Create two tables like below:

```

CREATE TABLE SKIPTABLE(SKID INT, SKNAME VARCHAR(30))
INSERT INTO SKIPTABLE VALUES (1, 'ABC')
INSERT INTO SKIPTABLE VALUES (2, 'MNO')
  
```

```

CREATE TABLE LKPTABLE(LKID INT, LKNAME VARCHAR(30))
INSERT INTO LKPTABLE VALUES (1, 'RIV')
INSERT INTO LKPTABLE VALUES (2, 'KLR')
  
```

E.g.: Text file abc ad mno klr Ad a ad ad Ad . afas  
mn cn ab mnk l op  
kk lon k rorp  
ppp king mmm lll ent

**BUSINESS SCENARIO:-**

TABLENAME	FILE COLUMNS	FILENAME	COLUMNNAME	TRANSFORMATION RULE	KEYVALUE
PARTY_HYD	USERNAME			USERNAME	
PARTY_HYD	PACKAGENAME			PACKAGENAME	
PARTY_HYD	ORGANIZATION	DATA_VARTEXT	ORGANIZATION	TRIM OF ORGANIZATION	
PARTY_HYD	PARTYCODE	DATA_VARTEXT	PARTYCODE	DIRECT MOVE	
PARTY_HYD	JDATE	DATA_VARTEXT	JDATE	DIRECT MOVE	
PARTY_HYD	PARTYINCOME	DATA_VARTEXT	PARTYINCOME	DIRECT MOVE	
<hr/>					
PARTY_MUM	USERNAME			USERNAME	
PARTY_MUM	PACKAGENAME			PACKAGENAME	
PARTY_MUM	ORGANIZATION	DATA_VARTEXT	ORGANIZATION	DIRECT MOVE	
PARTY_MUM	PARTYCODE	DATA_VARTEXT	PARTYCODE	DIRECT MOVE	
PARTY_MUM	JDATE	DATA_VARTEXT	JDATE	DIRECT MOVE	
PARTY_MUM	PARTYINCOME	DATA_VARTEXT	PARTYINCOME	DIRECT MOVE	
<hr/>					
PARTY_CHEENAI	USERNAME			USERNAME	
PARTY_CHEENAI	PACKAGENAME			PACKAGENAME	
PARTY_CHEENAI	ORGANIZATION	DATA_VARTEXT	ORGANIZATION	TRIM OF ORGANIZATION	
PARTY_CHEENAI	PARTYCODE	DATA_VARTEXT	PARTYCODE	DIRECT MOVE	
PARTY_CHEENAI	JDATE	DATA_VARTEXT	JDATE	DIRECT MOVE	
PARTY_CHEENAI	PARTYINCOME	DATA_VARTEXT	PARTYINCOME	DIRECT MOVE	
<hr/>					
LOCATIONS_SUMMARY	LOCATION	DATA_VARTEXT	LOCATION	DIRECT MOVE	
LOCATIONS_SUMMARY	SUM_INCOME	DATA_VARTEXT	See mapping Rule	Take sum of partyincome	
LOCATIONS_SUMMARY	AVG_INCOME	DATA_VARTEXT	See mapping Rule	Take avg of partyincome	
LOCATIONS_SUMMARY	MIN_INCOME	DATA_VARTEXT	See mapping Rule	Take min of partyincome	
LOCATIONS_SUMMARY	MAX_INCOME	DATA_VARTEXT	See mapping Rule	Take max of partyincome	

**BUSINESS SCENARIO:**

TABLENAME	FILE COLUMNS	FILENAME	COLUMNNAME	TRANSFORMATION RULE
PARTY_DETAILS	LAPARTYID	DATA_VARTEXT	PARTYID	DIRECT MOVE
PARTY_DETAILS	CITY	DATA_VARTEXT	PARTYNAME, PARTYLOC	PARTYNAME /First three characters of PARTYLOC

PARTY_DETAILS	INCOME	DATA_VARTEXT	See mapping rule	IF INCOME IS NULL DISPLAY 99999
PARTY_DETAILS	PARTYCODE DESC	DATA_VARTEXT	PARTYCODE	Use the below logic: PARTYCODEDESC="IT" if PARTYCODE=10 PARTYCODEDESC="HR" if PARTYCODE=20 FOR OTHER PARTYCODES THE PARTYCODEDESC IS 'SALES & MARKETING'
PARTY_DETAILS	JDATE	DATA_VARTEXT	JDATE	Display only month and year in the below format: 'yyyy-mm-dd'
PARTY_DETAILS	ORGANIZATION	DATA_VARTEXT	ORGANIZATION	Uppercase of Organization

**BUSINESS SCENARIO:**

TABLENAME	FILE COLUMNS	FILENAME	COLUMNNAME	TRANSFORMATION RULE	KEYVALUE
LOCATIONS	USERNAME	PARTY_HYD	USERNAME	DIRECT MOVE	
LOCATIONS	PACKAGENAME	PARTY_HYD	PACKAGENAME	DIRECT MOVE	
LOCATIONS	ORGANIZATION	PARTY_HYD	ORGANIZATION	DIRECT MOVE	
LOCATIONS	PARTYCODE	PARTY_HYD	PARTYCODE	DIRECT MOVE	
LOCATIONS	JDATE	PARTY_HYD	JDATE	DIRECT MOVE	
LOCATIONS	PARTYINCOME	PARTY_HYD	PARTYINCOME	DIRECT MOVE	
[Redacted]					
LOCATIONS	USERNAME	PARTY_MUM	USERNAME	DIRECT MOVE	
LOCATIONS	PACKAGENAME	PARTY_MUM	PACKAGENAME	DIRECT MOVE	
LOCATIONS	ORGANIZATION	PARTY_MUM	ORGANIZATION	DIRECT MOVE	
LOCATIONS	PARTYCODE	PARTY_MUM	PARTYCODE	DIRECT MOVE	
LOCATIONS	JDATE	PARTY_MUM	JDATE	DIRECT MOVE	
LOCATIONS	PARTYINCOME	PARTY_MUM	PARTYINCOME	DIRECT MOVE	
[Redacted]					
LOCATIONS	USERNAME	PARTY_CHENNAI	USERNAME	DIRECT MOVE	
LOCATIONS	PACKAGENAME	PARTY_CHENNAI	PACKAGENAME	DIRECT MOVE	
LOCATIONS	ORGANIZATION	PARTY_CHENNAI	ORGANIZATION	DIRECT MOVE	
LOCATIONS	PARTYCODE	PARTY_CHENNAI	PARTYCODE	DIRECT MOVE	
LOCATIONS	JDATE	PARTY_CHENNAI	JDATE	DIRECT MOVE	
LOCATIONS	PARTYINCOME	PARTY_CHENNAI	PARTYINCOME	DIRECT MOVE	

**BUSINESS SCENARIO:**

FILENAME	FILE COLUMNS	TABLENAME	COLUMNNAME	TRANSFORMATION RULE	KEYVALUE
PARTY_REPORT	LAPARTYID	PARTY	PARTYID	DIRECT MOVE	Y
PARTY_REPORT	USERNAME	PARTY	See mapping rule	Get USERNAME where LOCATIONS. PARTYCODE=PARTY.PARTYCODE	
PARTY_REPORT	PACKAGENAME	PARTY	See mapping rule	Get PACKAGE NAME where LOCATIONS. PARTYCODE=PARTY.PARTYCODE	
PARTY_REPORT	INCOME	PARTY	PARTYINCOME	DIRECT MOVE	
PARTY_REPORT	RATING	PARTY	PARTYINCOME	Use the below logic: Rating="Poor" if PINCOME <30000 Rating="Average" if PINCOME >30000 and PINCOME<50000 Rating="Good" if PINCOME >50000	
PARTY_REPORT	JDATE	PARTY	JDATE	JDATE FORMAT "DD-MM-YY"	
PARTY_REPORT	BUSINESS_DATE			CURRENT DATE	
PARTY_REPORT	ORG_SHORT	PARTY	ORGANIZATION	FIRST 3CHARS OF ORGANIZATION	

# Vinay Tech House

**BUSINESS SCENARIO**

TABLENAME	TABLE COLUMNS	FILENAME	COLUMNNAME	TRANSFORMATION RULE
PARTY_NORMALIZED	PARTYID	DATA_NORMALIZED	PARTYID	DIRECT MOVE
PARTY_NORMALIZED	PARTYNAME	DATA_NORMALIZED	PARTYNAME	PARTYNAME
PARTY_NORMALIZED	PARTYLOC	DATA_NORMALIZED	PARTYLOC	PARTYLOC
PARTY_NORMALIZED	SAVINGS	DATA_NORMALIZED	See mapping rule	Value of Savings for Partyid,Name,Loc combination
PARTY_NORMALIZED	CURRENT	DATA_NORMALIZED	See mapping rule	Value of Current for Partyid,Name,Loc combination
PARTY_NORMALIZED	DMAT	DATA_NORMALIZED	See mapping rule	Value of Dmat for Partyid,Name,Loc combination

**TABLES REQUIRED:**

```
CREATE TABLE PARTY5(PARTYID INTEGER,PARTYNAME VARCHAR(30),PARTYLOC
VARCHAR(30),USERCREATED VARCHAR(30))
```

```
INSERT INTO PARTY5(1,'VINAY','HYD');
```

```
CREATE TABLE PARTY6(PARTYID INTEGER,PARTYNAME VARCHAR(30),PARTYLOC
VARCHAR(30),STD_DT DATE,END_DT DATE)
```

```
INSERT INTO PARTY6(1,'VINAY','HYD','2011-01-19','0001-01-01');
```

**BUSINESS SCENARIO: FIXED WIDTH FORMAT (SCD TYPE1)**

BDW TABLENAME	BDW COLUMN NAME	FILENAME	COLUMNNAME	Mapping REMARKS	Transformation Rule
PARTYS	Party Id	DATA_FIXEDTEXT	PARTYID	PARTYID	DIRECT MOVE
PARTYS	Partyname	DATA_FIXEDTEXT	Partyname	PARTYNAME	DIRECT MOVE
PARTYS	PARTYLOC	DATA_FIXEDTEXT	PARTYLOC	See mapping rule	if partyid exists location should be modified else insert
PARTYS	USERCREATED	DATA_FIXEDTEXT		See mapping rule	DEFAULT USER

**BUSINESS SCENARIO: VARIABLE TEXT FORMAT (SCD TYPE2, EFFECTIVE DATE RANGE)**

BDW TABLENAME	BDW COLUMN NAME	FILENAME	COLUMNNAME	Mapping REMARKS	Transformation Rule
PARTY6	Party Id	DATA_VARTEXT	PARTYID	PARTYID	DIRECT MOVE
PARTY6	Partyname	DATA_VARTEXT	Partyname	PARTYNAME	DIRECT MOVE
PARTY6	PARTYLOC	DATA_VARTEXT	PARTYLOC	See mapping rule	DIRECT MOVE
PARTY6	ST_DT	DATA_VARTEXT	ST_DT	ST_DT	DIRECT MOVE
PARTY6	END_DT	DATA_VARTEXT		See mapping rule	1.If new record came inserted with start_dt and end date as '0001-01-01' 2. If partyid existed a.should be inserted st_dt and end date as '0001-01-01' b.Earlier record end date

				should be new record start date
--	--	--	--	------------------------------------

**TABLES REQUIRED:**

```
CREATE TABLE PARTY7(PARTYID INTEGER,PARTYNAME VARCHAR(30),PARTYCODE INTEGER)
```

```
INSERT INTO PARTY7(1,'KKKKK',20)
```

```
CREATE TABLE PARTY8(PARTYID INTEGER,PARTYNAME VARCHAR(30),PARTYCODE INTEGER,STATUS VARCHAR(30))
```

```
INSERT INTO PARTY8(1,'VINAY','HYD')
```

**BUSINESS SCENARIO: DATA IN VARIABLE TEXT FORMAT**

BDW TABLENAME	BDW COLUMN NAME	FILENAME	COLUMNNAME	Mapping REMARKS	Transformation Rule
					1.Delete the record if transaction code is 'D' 2.Do upsert operation if the transaction code is 'U'
PARTY7	Party Id	DATA_VARTEXT	PARTYID	PARTYID	PARTYID
PARTY7	Partyname	DATA_VARTEXT	Partyname	See mapping rule	If partyid exists update it else insert it
PARTY7	Partycode	DATA_VARTEXT	PARTYCODE	See mapping rule	<u>If partyid exists update it else insert it</u>

**BUSINESS SCENARIO: VARIABLE TEXT FORMAT (SCD TYPE 2 CURRENT EXPIRED MECHANISM)**

BDW TABLENAME	BDW COLUMN NAME	FILENAME	COLUMNNAME	Mapping REMARKS	Transformation Rule
PARTY8	Party Id	DATA_VARTEXT	PARTYID	PARTYID	DIRECT MOVE
PARTY8	Partyname	DATA_VARTEXT	Partyname	PARTYNAME	DIRECT MOVE
PARTY8	PARTYLOC	DATA_VARTEXT	PARTYLOC	See mapping	DIRECT MOVE

				rule	
PARTY8	STATUS	DATA_VARTEXT	STATUS	See mapping rule	1. If new record came inserted with STATUS 'Current' 2. If partyid existed a. should be inserted with Status as 'Current' b. Earlier record Status should be 'Expired'

### CONTAINER

\* Diff. b/w group and container.

- GROUP cannot have any prop, it is a logical separation. whereas CONTAINER contain properties.

Containers are objects that help SSIS provide structure to one or more tasks. They can help you look through a set of tasks until a criteria has been met. It is a logical separation. whereas CONTAINER contain properties.

### TASK HOST

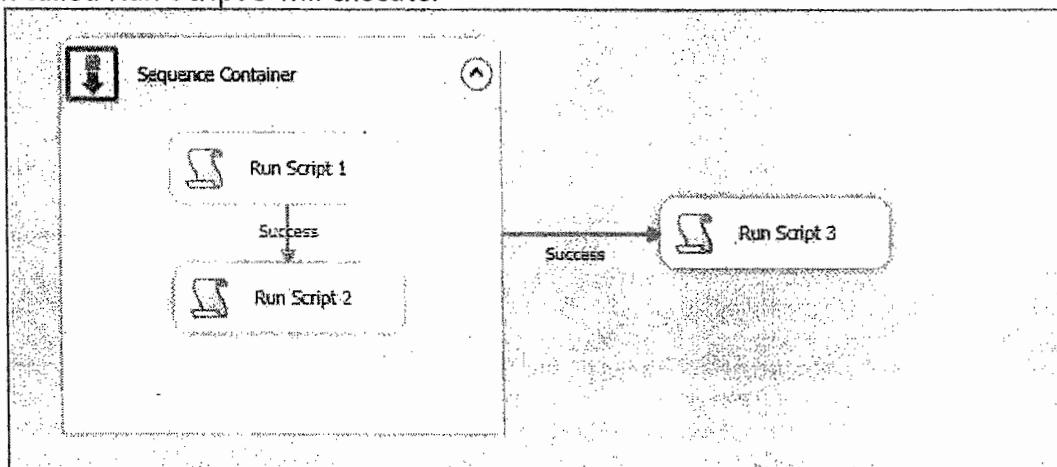
The Task Host Container is the default container under which single tasks fall and is used only in the background scenes for SSIS. You won't notice this type of container. You'll notice that this type of container is not in the Toolbox in Visual Studio and is implicitly assigned to each task. In fact, even if you don't specify a container for a task, it will be placed in a Task Host Container. The SSIS architecture extends variables and event handlers to the task through the Task Host Container.

### SEQUENCE CONTAINER

Sequence Containers handle the flow of a subset of a package and can help you divide a pkg into smaller and more manageable pieces. Some nice applications that you can use Sequence Containers for include the following:

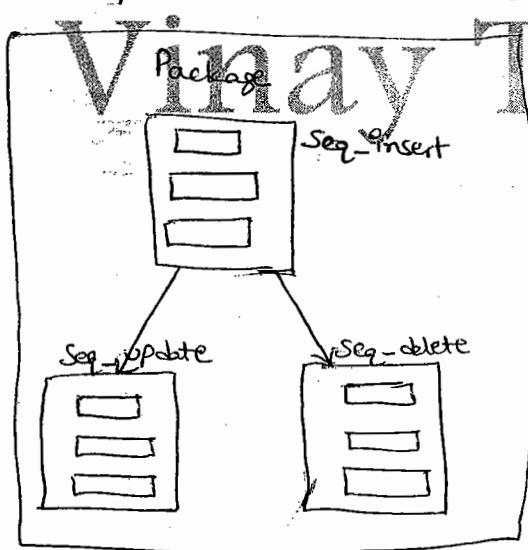
- Grouping tasks so that you can disable a part of the package that's no longer needed
- Narrowing the scope of the variable to a container
- Managing the properties of multiple tasks in one step by setting the properties of the container
- Using one method to ensure that multiple tasks have to execute successfully before the next task executes
- Creating a transaction across a series of data-related tasks, but not on the entire package
- Creating event handlers on a single container, wherein you could send an e-mail if anything inside one container fails and perhaps page if anything else fails.

shows an example of a Sequence Container in which two tasks must execute successfully before the task called Run Script 3 will execute.



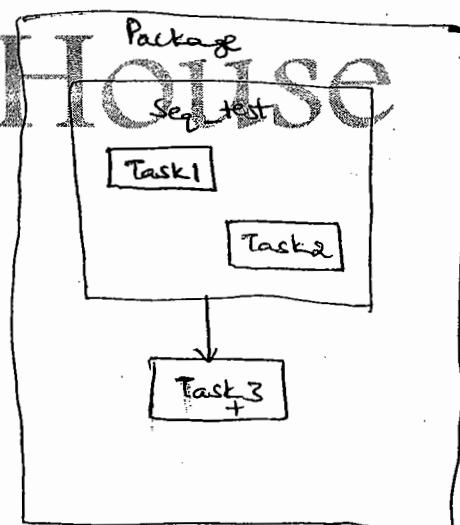
### Transactional operation

- After 3 successful inserts 3 successful updates should happen, any single insert fails perform 3 successful deletes.



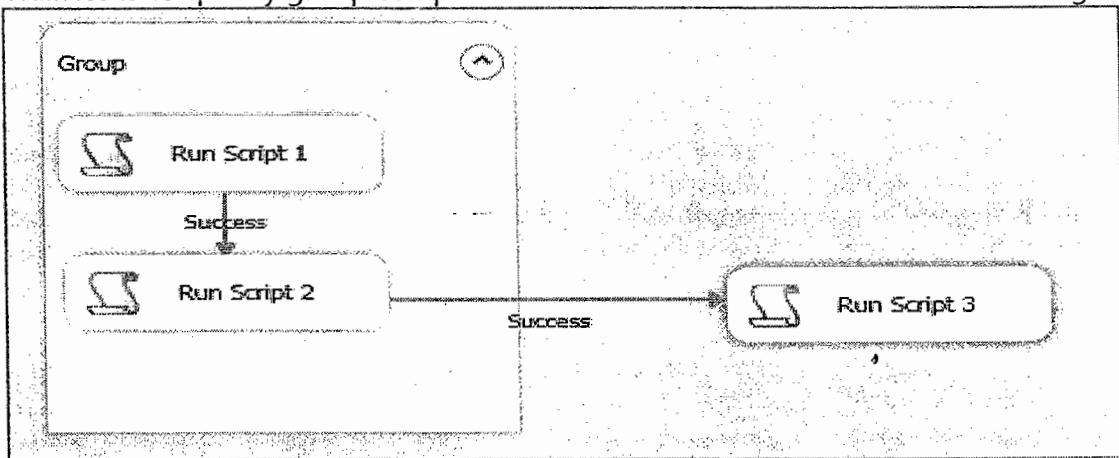
### Parallel processing

- Execute Task1 and Task2 parallelly, later after finishing both execute Task3.



## GROUPS

Groups are not actually containers but simply a way to group components. A key difference between groups and containers is that pkg cannot be deleted through a container. Because of this, groups don't have precedence constraints originating from them (only from the tasks). And you cannot disable the entire group, as you can with a Sequence Container. Their only usefulness is to quickly group components in either a Control Flow or a Data Flow together.



## FOR LOOP CONTAINER

# Vinay Tech House

The For Loop Container enables you to create looping in your pkg similar to how you would loop nearly in any prog. language. In this looping style, SSIS optionally initializes an expression and continues to evaluate it until the expression evaluates to false.

The following simple example demonstrates the functionality of the For Loop Container, whereby you'll use the container to loop over a series of tasks five times. Although this example is rudimentary, you can plug in whatever task you want in place of the Script Task.

```
Public Sub Main()
```

```
' Add your code here
```

```
MessageBox.Show(Dts.Variables("Counter").Value)
```

```
Dts.TaskResult = ScriptResults.Success
```

```
End Sub
```

Diff b/w For loop and for each loop?

### For Loop

1. We know end point.  
(no. of iterations we know)

2. Variable required

Eg.: Loading the below file 3 times.

- Loading 3 work sheets in an excel Three tables.

### For each loop

1. We don't know the end point.  
(Every object till object ends).
2. Enumerator (structure or object).

Eg.: All files in a folder loading to a table.

All datasets load to a table.

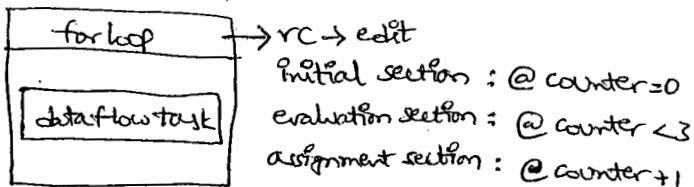
Eg: load data from a file into table & timer.

- Create DataFlow with FlatFile and OLEDB destination.
- Create a variable named counter.

SSIS Menu → Variables

click Add variable → Name it as counter  
Type as int

- Use Dataflow and Variable like below.



- Run the pkg and observe the result.

Exercises: Load three work sheets in an excel sheet to a table.  
Load three work sheets in an excel to three tables.

Refer to script &  
examples for  
answers.

## FOR EACH LOOP CONTAINER

# Vinay Tech House

The Foreach Loop Container is a powerful ~~looping mechanism that enables you to loop through a collection of objects~~. As you loop through the collection, the container assigns the value from the collection to a variable, which can later be used by tasks or connections inside or outside the container. You can also map the value to a variable. The types of objects that you will loop through vary based on the enumerator you set in the editor in the Collection page.

The properties of the editor vary widely according to what you set for this option:

**Foreach File Enumerator:** Performs an action for each file in a directory with a given file extension

**Foreach Item Enumerator:** Loops through a list of items that are set manually in the container

**Foreach ADO Enumerator:** Loops through a list of tables or rows in a table from an ADO recordset

**Foreach ADO.NET Schema Rowset Enumerator:** Loops through an ADO.NET schema

**Foreach From Variable Enumerator:** Loops through an SSIS variable

**Foreach Nodelist Enumerator:** Loops through a node list in an XML document

**Foreach SMO Enumerator:** Enumerates a list of SQL Management Objects (SMO)

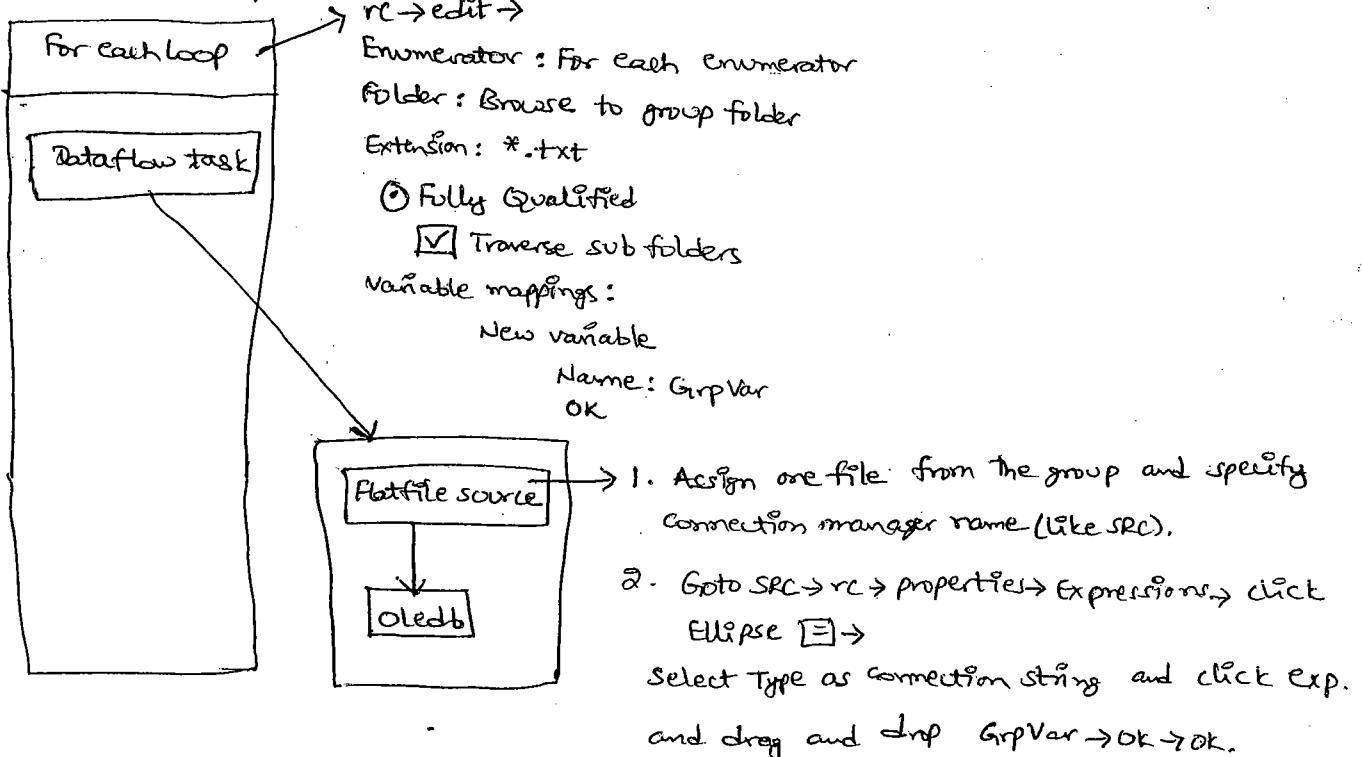
The most important of the enumerators is the Foreach File enumerator since it's used more frequently. In this next example, you'll see how to loop over a number of files and perform an action on each file. The second most important enumerator is the Foreach ADO enumerator, which loops over records in a table.

### Foreach ADO Enumerator Example

The Foreach ADO Enumerator loops through a collection of records and will execute anything inside the container for each row that is found.

Loading set of files with the same structure to a table.

1. Take Foreach Loop



# Vinay Tech House

## FILE SYSTEM TASK

The File System Task is a configurable GUI component that performs file operations available in the `System.IO.File` class. If you are used to coding in VBScript, this is an out-of-the-box replacement for the VBScript utility classes that you used to write using the COM-based

FileSystemObject. In either case, the File System Task can perform basic file operations such as the following:

**Copy Directory:** Copies all files from one directory to another. You must provide the source and destination directories.

**Copy File:** Copies a specific file. You must provide the source and destination filename.

**Create Directory:** Creates a directory. You must provide the source directory name and indicate whether the task should fail if the destination directory already exists.

**Delete Directory:** Deletes a directory. You must provide the source directory to delete.

**Delete Directory Content:** Deletes all files in a source directory.

**Delete File:** Deletes a specifically provided source file.

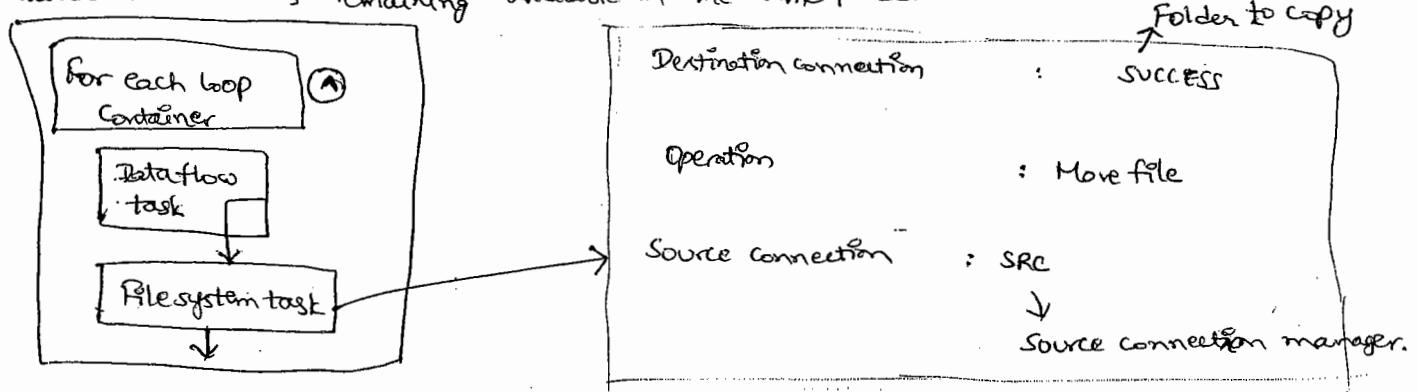
**Move Directory:** Moves a provided source directory to a destination directory. You must indicate whether the task should fail if the destination directory already exists.

**Move File:** Moves a specific provided source file to a destination. You must indicate whether the task should fail if the destination file already exists.

**Rename File:** Moves a specific provided source file to a destination by changing the name. You must indicate whether the task should fail if the destination file already exists.

**Set Attributes:** Sets Hidden, Read-Only, Archive, or System attributes on a provided source file.

Eg: In Above foreach loop load successful files into a folder called SUCCESS so that failed file onwards remaining available in the same place.



### VARIOUS WAYS OF DEBUGGING

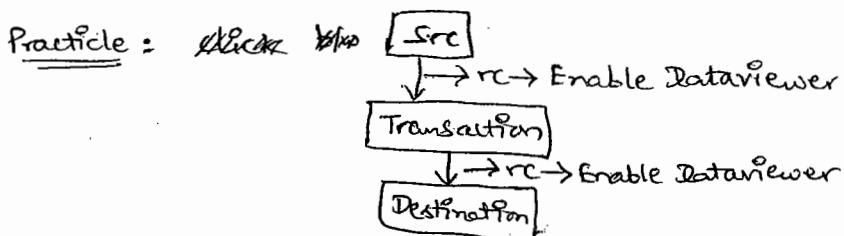
A good debugger is a good developer.

Debugging is a process to identify errors and issues in detail.

There are 3 ways to debug. a) Partial execution - There are many tasks in a pkg and you want to start a particular task for observing the operation this partial executions helpful.

**Vinay Tech House**

b) Dataflow level data viewers: Data viewers display the data to various components such as source, transformation and destination.



c) Breakpoints and Hit counts: We take a control flow level to ~~select~~ to take a break at a particular event (on success, on failure etc).

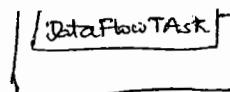
Eg: Take a break after execute sql task successful operation.

rc → Execute SQL task → edit breakpoints → on post execute → always. 0

Hit count: To reduce the no. of breakpoints when operating inside a loop we take hit count.

- FOR LOOP CONTAINER → RC → edit →
  $\begin{array}{l} @i=1 \\ @i<10 \\ @i = @i + 1 \end{array}$

## MSBI 2012 (SQL Server Integration Services)



Hit count equals : 2  
Hit count greater than equals : 2  
Hit count multiple : 2

Hit count Type → The type what you are selecting

Hit count → Value what is has to implement

Hit count Types

a) Hit count Equals

Once it reaches hit count value it takes a break.

b) Hit count greater than or equals

From the specified value onwards everytime time it takes a break.

c) Hit count multiple

Break occurs at multiples of the hit count value.

Ex: Hit count multiple 3 → Every 3rd occurrence there will be a break.

## SSAS PROCESSING

→ SSAS operations in SSIS.

Processing SSAS objects in SSIS can be as easy as using the Analysis Services Processing Task. However, if your SSAS cubes require adding or processing specific partitions or changing the names of cubes or servers, then you will need to consider other approaches. In fact, many, if not most, solutions require using other processing methods. SSAS in SQL Server 2012 has two types of models, multidimensional and tabular. Both of these models require processing. For multidimensional models, you are processing dimensions and cube partitions. For tabular models, you are processing tables and partitions. However, both models have similar processing options.

The primary ways to process SSAS models through SSIS include the following:

- Analysis Services Processing Task:** Can be defined with a unique list of dimensions, tables, and partitions to process. However, this task does not allow modifications of the objects through expressions or configurations.
- Analysis Services Execute DDL Task:** Can process objects through XMLA scripts. The advantage of this task is the capability to make the script dynamic by changing the script contents before it is executed.
- Script Task:** Can use the API for SSAS, which is called AMO (or Analysis Management Objects). With AMO, you can create objects, copy objects, process objects, and so on.
- Execute Process Task:** Can run ascmd.exe, which is the SSAS command-line tool that can run XMLA, MDX, and DMX queries. The advantage of the ascmd.exe tool is the capability to pass in parameters to a script that is run.

## ANALYSIS SERVICES TASKS

Three tasks can be used for Analysis Services in SSIS: the Analysis Services Execute DDL Task, the Processing Task, and the Data Mining Task.

**Analysis Services Execute DDL Task** - To create, manipulate or remove cube objects this task is helpful. (Cube, dimension, fact etc).

The SQL Server Analysis Services Execute DDL Task is the Analysis Services equivalent of the Execute SQL Task, but it is limited in scope to issuing Data Definition Language statements. The task simply executes a DDL statement against an Analysis Services system. Typically, you would use DDL statements to create a cube, a dimension, or any other online analytical processing (OLAP) object.

You can easily find XMLA statements by going to Management Studio and scripting a given action. In the example just mentioned, you can open Management Studio and delete the dimension, but instead of clicking OK, click the Script icon.

You can easily find XMLA statements by going to Management Studio and scripting a given action. In the example just mentioned, you can open Management Studio and delete the dimension, but instead of clicking OK, click the Script icon.

**Analysis Services Processing Task** - It deals with data by loading to the objects.

The SQL Server Analysis Services Processing Task takes care of the processing of Analysis Services objects. If you are familiar with using the Analysis Service projects in SSDT, then you'll be familiar with the task of processing a cube, dimension, or mining object.

Navigation: Edit → Specify analysis service server name cube name in the connection mgr  
click add and specify dimension and fact table name → OK.

### DATA MINING QUERY TASK

The Data Mining Query Task enables you to run predictive queries against your Analysis Services data-mining models and output the results to a data source. The Data Mining Query Task is similar to the Analysis Service Execute DDL Task in that you can execute subsequent mining queries against a processed mining model in Analysis Server.

Vinay Tech House

### DATA PREPARATION TASKS

Before processing data from other systems, you sometimes have to first retrieve it or validate the content to determine your level of confidence in the data's quality. SSIS provides a set of tasks that can be used to retrieve data files using the files and folders available in the file system, or it can reach out using FTP and web service protocols.

### DATA PROFILER

- Screening the data before load to destination.
- It comes under Data Quality Process.
- Advance settings enhancement available in DQS.

Data profiling is the process of examining data and collecting metadata about the quality of data about frequency, statistical patterns, interdependencies, uniqueness and redundancy. This type of analytical activity is important for the overall quality and health of an operational data store (ODS) or data warehouse. The following list describes the different request types and how you can use them to profile your data:

- a) **Candidate Key Profile Request:** The profile request will examine a column or set of columns to determine the likelihood of there being a unique candidate key for the data set. Use this to determine whether you have duplicate key values or whether it is possible to build a natural key with the data.
- b) **Column Length Distribution Profile:** This profile request enables you to analyze the statistical profile of all the data in a column, with the percentage of incidence for each length. You can use

this to help you determine whether your data column length settings are set correctly or to look for bad data in attributes that are known to be one fixed size.

- c) **Column Null Ratio Profile Request:** This profile request looks at the ratio of NULL values in a column. Use this to determine whether you have a data quality problem in your source system for critical data elements.
- d) **Column Pattern Profile Request:** This profile request enables you to apply regular expressions to a string column to determine the pass/fail ratio across all the rows. Use this to evaluate business data using business formatting rules.
- e) **Column Statistics Profile Request:** This profile request can analyze all the rows and provide statistical information about the unique values across the entire source. This can help you find low incidence values that may indicate bad data. For example, a finding of only one color type in a set of 1 million rows may indicate that you have a bad color attribute value.
- f) **Functional Dependency Profile Request:** This is one of two profile requests that enable you to examine relationships between tables and columns to look for discrepancies within a known dependency. For example, you can use this request to find countries with incorrect currency codes.
- g) **Value Inclusion Profile Request:** This profile request tests to determine whether the values in one column are all included in a separate lookup or dimension table. Use this to test foreign key relationships.

NOTE: 1. The drawback with this task is rules are already defined. (we cannot add our requirements for quality check)

2. This profiler gives an dp file with statistics, we can open through data profiler manually for observation OR by clicking data profile viewer in dataflow task itself.

Analyzing the below requirements on the data

- a) No name should be null
- b) No dept code should be more than 2 digits
- c) Income should be b/w 10000 and 50000 etc--

1. Take data profile Task →

General →

Destination connection: specify file to send Data profile results.

click quick profile button → & specify database and Table to profile → OK

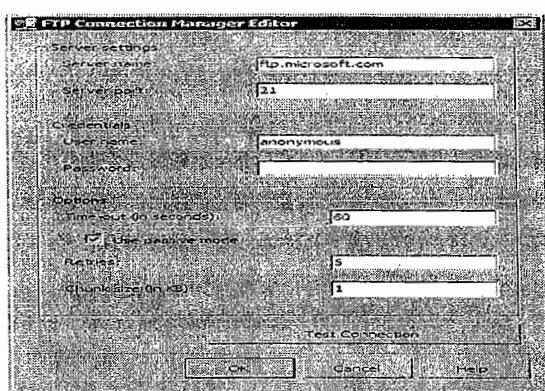
2. Run the data profile Task observe the profile created in the specified folder.

Analyzing the profiled file: Two ways.

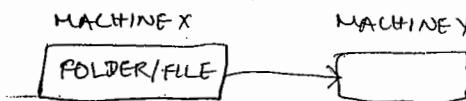
### FTP TASK

- a) Running mode do data profile task → click data.profile.viewer → Monitor statistics
- b) start → programs → Sqlserver 2012 → Integration services → Data profile viewer → Open file and specify file, monitor statistics.

The SSIS FTP Task enables the use of the File Transfer Protocol (FTP) in your package development tasks. This task now exposes more FTP command capability, enabling you to create or remove local and remote directories and files. Another change from the legacy DTS FTP Task is the capability to use FTP in passive mode. This solves the problem that DTS had in communicating with FTP servers when the firewalls filtered the incoming data port connection to the server.



REAL TIME: Moving files/folders from one server to other (Filesystem to other).



1. FTP Server name
2. File system i) servername  
folder/files.  
operation: copy/move
3. File system ii) servername  
folder/files.

The General tab in the FTP Task Editor is where you specify the FTP Connection Manager for the FTP site you wish to access. If you haven't specified one, follow these steps:

1. Select <New Connection...> under the FTPConnection property. This will open the FTP Connection Manager, where you can configure the FTP connection. In Figure 3-11, the Server Name property contains the FTP address for the FTP server. The Server Port property is set to 21, which is the default port for most FTP sites. You can change this if necessary. The other important option to note here is the "Use passive mode" checkbox, which is new to SSIS.
2. Once you have the FTP connection configured, move to the File Transfer tab. The IsRemotePathVariable and IsLocalPathVariable properties allow the paths to be set to an optional variable. Using variables enables you to set these values dynamically at runtime. The RemotePath property sets the directory or files for the remote FTP system.
3. Once the FTPConnection property from the General tab has been selected, you can browse to the actual remote file system to select the remote path or file by clicking the ellipsis in the Remote Path property. You'll see a dialog similar to the one shown in Figure 3-12 for browsing the FTP remote paths (and files if you choose Receive files for the Operation property).

## WEB SERVICE TASK

→ It calls and executes web services

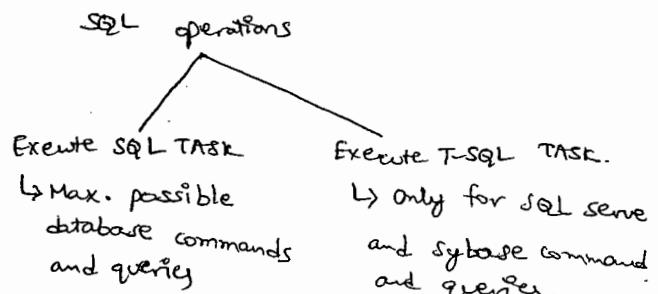
The Web Service Task in SSIS is used to retrieve XML-based result sets by executing a method on a web service. Web services are a big part of advancing service-oriented architectures, and they can be used in SSIS to provide real-time validation of data in your ETL processes or to maintain lookup or dimensional data.

## RDBMS SERVER TASKS

### Bulk Insert Task (See the initial examples)

### EXECUTE SQL TASK

The Execute SQL Task is one of the most widely used tasks in SSIS for interacting with an RDBMS Data Source. The Execute SQL Task is used for all sorts of things, including truncating a staging table data prior to importing, retrieving row count to determine the next step in a work flow (or) calling stored procedure to perform business logic against staged data. This task is also used to retrieve information from a database repository.



## EXECUTING A PARAMETERIZED SQL STATEMENT

The task can execute a SQL command in two basic ways:

by executing inline SQL statements (Direct Statement Writing) or by executing stored procedures.

UPDATE DimProduct Set Color = 'Red' Where ProductKey = ?

Here the parameter mapping collection maps the first parameter [ordinal position of zero (0)] to a user variable. When mapping parameters to connections and underlying providers, use the following table to set up this tab in the Task Editor:

IF USING CONNECTION OF TYPE	PARAMETER MARKER TO USE	PARAMETER NAME TO USE
ADO	?	Param1, Param2, ...
ADO.NET	@<Real Param Name>	@<Real Param Name>
✓ ODBC	?	1, 2, 3 (note ordinal starts at 1)
✓ OLEDB and EXCEL	?	0, 1, 2, 3 (note ordinal starts at 0)

Note:

*Building inline dynamic SQL statements with variables no longer has a limitation of 4,000 characters as it did with SQL Server 2005 and 2008.*

## EXECUTING A BATCH OF SQL STATEMENTS

Use these rules as a guide for executing a batch of SQL statements:

Use GO statements between each distinct command. Note that some providers allow you to use the semicolon (;) as a command delimiter. If there are multiple parameterized statements in the batch, all parameters must match in type and order. Only one statement can return a result, and it must be the first statement. If the batch returns a result, then the columns must match the same number and properly named result columns for the Execute SQL Task. If the two don't match and you have subsequent UPDATE or DELETE statements in the batch, these will execute even though the results don't bind, and an error results. The batch is sent to SQL Server to execute and behaves the same way.

## CAPTURING SINGLETON RESULTS

On the General tab of the Execute SQL Task, you can set up the task to capture the type of result that you expect to have returned by configuring the ResultSet property. This property can be set to return nothing, or None, a singleton result set, a multi-line result, or an XML-formatted string. Any setting other than None requires configuration of the Result Set tab on the editor. In the Result Set tab, you are defining the binding of returned values into a finite set of SSIS variables.

```
SELECT TOP 1 CarrierTrackingNumber, LineTotal, OrderQty, UnitPrice
From Sales.SalesOrderDetail;
```

To capture these columns from this table, you need to create some variables in the package.

## MULTIROW RESULTS

Typically, you capture multi-row results from a database as a recordset or an XML file (particularly between SQL Server Data Sources) to use in another Script Task for analysis or decision-making purposes, to provide an enumerator in a Foreach or Looping Task, or to feed into a Data Flow Task for processing. Set up the SQLSourceType and SQLStatement properties to call either an inline SQL statement or a stored procedure. In either case, you would set the ResultSet property in the General tab to Full Result Set, and the Result Set tab is set up to capture the results. The only difference from capturing a singleton result is that you need to capture the entire result into a variable, rather than map each column. The recordset can only be captured in a variable with the object data type. Once the recordset is stored as a variable, you can do things like "shred" the recordset. The term *shredding* means iterating through the recordset one row at a time in a Foreach Loop operation. For each iteration, you can capture the variables from, and perform an operation on, each row transform. To do this, just create a Source Script Transform in a Data Flow and add to it the columns that you want to realize from the stored recordset and pass in the recordset variable. Then add code similar to the following to turn the column data from the recordset into the output stream (to save time and space, only two columns are being realized in the recordset):

VB

```
Public Overrides Sub CreateNewOutputRows()
    Dim oleDA As New System.Data.OleDb.OleDbDataAdapter()
    Dim dT As New System.Data.DataTable()
    Dim row As System.Data DataRow
    oleDA.Fill(dt, Variables.RecordSetResult)
    For Each row1 In dt.Rows
        Output0Buffer.AddRow()
        Output0Buffer.CarrierTrackingNumber = _
            row("CarrierTrackingNumber").ToString()
        Output0Buffer.UnitPrice = System.Convert.ToDecimal(row(6))
    Next
End Sub
```

The XML version of capturing the result in a string is even easier. You don't need to use the Script Component to turn the XML string back into a source of data. Instead, use the out-of-the-box component called the XML Source in the Data Flow. It can accept a variable as the source of the data.

## EXECUTING A STORED PROCEDURE

The table provides a reference to how you should code the SQLStatement property in the Execute SQL Task:

IF USING CONNECTION TYPE	AND ISQUERYSTOREDPROCEDURE	CODE THE SQL STATEMENT PROPERTY LIKE THIS
OLEDB and EXCEL	N/A	EXEC usp_StoredProc ?, ?
ODBC	N/A	(call usp_StoredProc (?, ?))
ADO	false	EXEC usp_StoredProc ?, ?
	true	usp_StoredProc
ADO.NET	false	EXEC usp_StoredProc @Parm1, @Parm2
	true	usp_StoredProc @Parm1, @Parm2

## RETRIEVING OUTPUT PARAMETERS FROM A PROCEDURE

The main thing to remember is that all retrieved output or return parameters have to be pushed into variables to have any downstream use.

To get the defined SQL Server data type parameters to match the SSIS variables, you need to set up the parameters with these mappings:

PARAMETER NAME	SQl SERVER DATA TYPE	SSIS DATA TYPE
@CARRIER_TRACKING_NUMBER	nvarchar	string
@LINE_TOTAL	numeric	double
@ORDER_QTY	smallint	int16
@UNIT_PRICE	money	double

## PRACTICALS FOR THE ABOVE ALL SCENARIOS IN EXECUTE SQL

1. Executing single statement

Connection Type : OLEDB  
 Connection : VINAYAKA\_DB  
 SQL Statement : delete from TAB where tid = 3;

2. Executing Multiple statements

SQL Statement : go  
 DELETE FROM TAB WHERE TID = 3;  
 go  
 delete from emp\_stones;

3. SQL statement with parameters

a) Create two parameters ID (INT), NM (String) and assign values 10, RR

b) SQL Statement : Insert into TAB values (?,?)

c) Parameters

add → parameter dt parameter name  
 user: ID LONG 0  
 user: NM varchar 1.

4. Execute SQL Task :

General :

Result Set : Single Row

SQL Statement : select top 1 tid, tnm from tab;

Result Set

Result	Parameter
tid	user: ID
tnm	user: NM

Script Task

Select C#

Randomly Variables : user: ID, user: NM

Edit Script : write the below in main function

MessageBox.Show("Employee ID " + Dts.Variables["ID"].Value + " Employee Name " + Dts.Variables["NM"].Value);

CREATING PROCEDURE with no params and working with it

DATABASE ENGINE → DB\_MSBI → CREATE PROCEDURE PR

```
AS
BEGIN
    INSERT INTO TAB VALUES(100,'xx');
    EXECUTE SQL TASK;
```

CREATING PROCEDURE with params and working with it

CREATE PROCEDURE PRI (@pid int, @pnm varchar(30))

AS

BEGIN

INSERT INTO TAB VALUES(@pid, @pnm);

END;

Execute sql task;

Sql statement; Exec PRI ?, ?;

parameters

variable	datatype	parameter name
----------	----------	----------------

ID	Long	0
NM	varchar	1

PROCEDURE TO RETURN VALUE:

1. Create procedure like below

CREATE PROCEDURE PR\_EMP (@ID INT, @NM VARCHAR(30) OUT)

AS BEGIN SELECT @NM=ENAME FROM EMP WHERE Eid=@ID; END;

Calling:

DECLARE @NM VARCHAR(30);

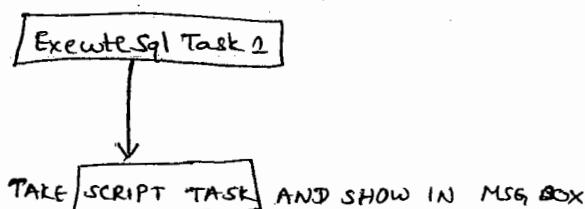
EXEC PR\_EMP 3, @NM OUTPUT;

PRINT @NM;

2. Goto execute package and implement the below steps

- a) Create variable NM of TYPE STRING

- b) Take two execute sql tasks like below and set the properties.



Connection: DB\_MSBI

Sql statement Exec PR\_EMP 3, ? OUTPUT

Parameters:

Name	Direction	Type	Parameter
NM	OUTPUT	varchar	0

## WORK FLOW TASKS

### EXECUTE PACKAGE TASK

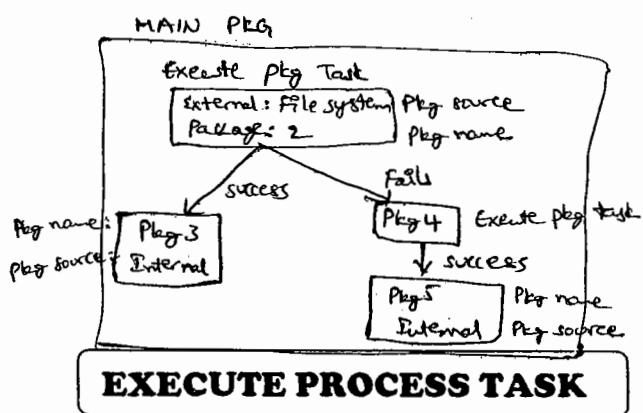
The Execute Package Task enables you to build SSIS solutions called parent packages that execute other packages called child packages.

**The child packages can be run as either in-process or out-of-process executables.** In the Package tab of the Execute Package Task Editor is the ExecuteOutOfProcess property; set to the default value of false, it will execute the package in its own process and memory space. **The task enables you to easily map parameters in the parent package to the child packages now too.** The ReferenceType option can be either External or Project References. This means you can point to a package inside your current project or outside the project to a SQL Server or file system. The best (easiest) option is to refer to a package in a project, as this option will easily "repoint" the reference as you migrate to production.

Next, go to the Parameter Bindings tab to pass parameters into the child package. First, select any parameters in the child package from its dropdown box, and then map them to a parameter or variable in the parent package.

General	Package		External Reference
<b>Package</b>	ReferenceType	Location	SQL Server
Parameter bindings	Connection	PackageName	*****
Expressions	Password	ExecuteOutOfProcess	False

Run pkg3 if pkg 2 successful , If fails run pkg 4. Pkg 4 success run pkg5



External: NOT IN THE PROJECT

(File system, sql server storage etc..)

Internal: Inside the project (2012 introduced)

\* Use execute out of process ~~task~~ pkg  
 functionality is simple, if it is take true pkg  
 functionality complex.

false

The Execute Process Task will execute a Windows or console application inside of the Control Flow. The most common example would have to be unzipping packed or encrypted data files with a command-line tool.

- To un-zip the folders
- To install softwares
- To execute DLL etc.

**MESSAGE QUEUE TASK**

→ Send (or) receive messages by keeping in queue.

The Message Queue Task enables you to send or receive messages from Microsoft Message Queuing (MSMQ) right out of the box. For integration with other messaging systems like IBM's MQ Series or Tibco's Rendezvous, you need to either code to a library within a Script Task, create a custom component, or execute T-SQL statements to a SQL Server Service Broker queue. Messaging architectures are created to ensure reliable communication between two disparate subsystems.

**SEND MAIL TASK**

→ Sending E-mails.

The Send Mail Task provides a configurable SSIS task for sending e-mail messages via SMTP. In legacy DTS packages, you had to send messages out through MAPI, which meant installing Outlook on the server on which the package was running. That is now no longer a requirement. Most of the configuration options are set in the Mail tab of the Send Mail Task Editor.

It requires SMTP server for mails delivery.

**A Mail**

SMTP Connection	→ 127.0.0.1
From	→ viraytech@gmail.com
To	→ madhu@gmail.com
Cc	
Bcc	
Subject	
Message source Type	→ Direct IP
Attachments	→ EDW-S.kml

**WMI DATA READER TASK**

# Vinay Tech House

Windows Management Instrumentation (WMI) is one of the best-kept secrets in Windows. WMI enables you to manage Windows servers and workstations through a scripting interface similar to running a T-SQL query. The WMI Data Reader Task enables you to interface with this environment by writing WQL queries (the query language for WMI) against the server or workstation (to look at the Application event log, for example). The output of this query can be written to a file or variable for later consumption.

Following are some applications for which you could use the WMI Data Reader Task:

- ✓ Read the event log looking for a given error.
  - ✓ Query the list of applications that are running.
  - ✓ Query to see how much RAM is available at package execution for debugging.
  - ✓ Determine the amount of free space on a hard drive.
- } Perform wind  
data reading  
operations.

**WMI EVENT WATCHER TASK**

→ It watcher window events.

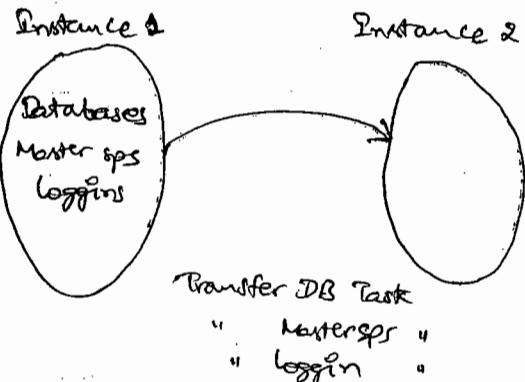
The WMIEvent Watcher Task empowers SSIS to wait for and respond to certain WMIevents that occur in the operating system. The task operates in much the same way as the WMI Data Reader Task operates. The following are some of the useful things you can do with this task:

- Watch a directory for a certain file to be written.
- Wait for a given service to start.
- Wait for the memory of a server to reach a certain level before executing the rest of the package or before transferring files to the server.
- Watch for the CPU to be free.

Realtime situation: Run The pkg once The required files available in the specified folder.

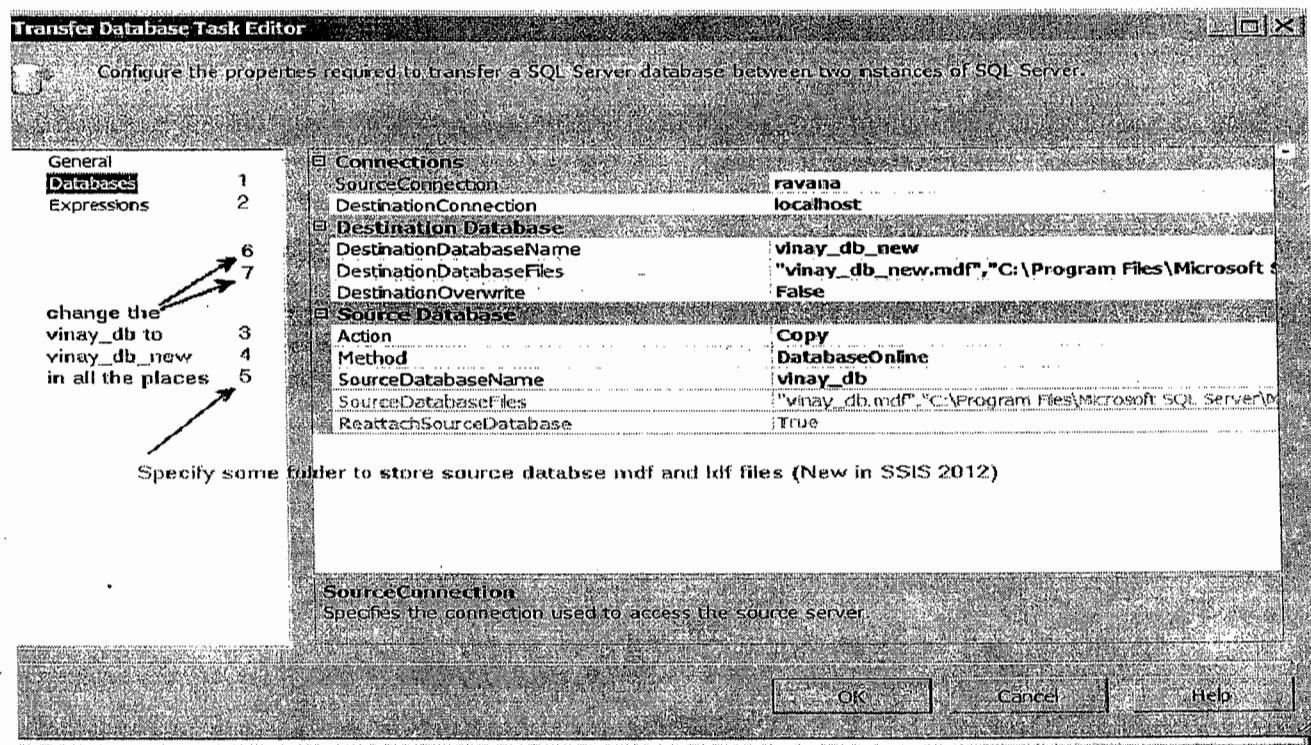
## TRANSFER TASKS

-  Transfer Database Task
-  Transfer Error Messages Task
-  Transfer Jobs Task
-  Transfer Logins Task
-  Transfer Master Stored Procedures Task
-  Transfer SQL Server Objects Task



## TRANSFER DATABASE TASK

The Transfer Database Task has, as you would expect, a source and destination connection and a database property. The other properties address how the transfer should take place.



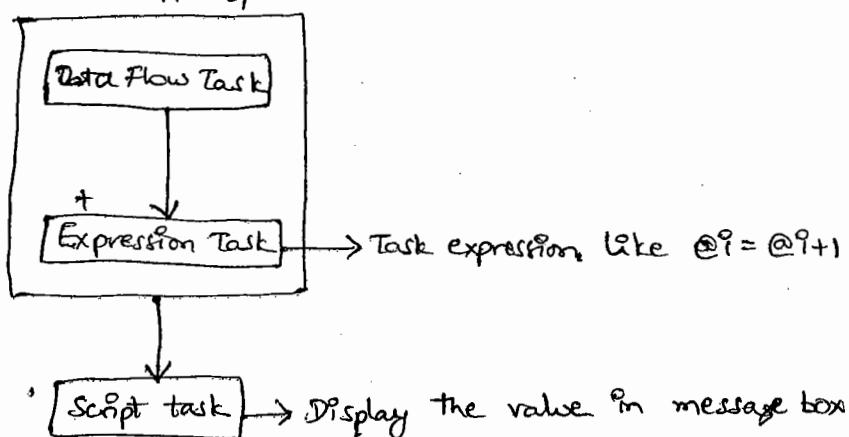
## CDC CONTROL TASK

**EXPRESSION TASK**

→ This is introduced in 2014 as part of value assignment to variables. In the older versions if you do this script task can evaluate an expression property required. In the latest version it is simplified by using expression task.

Below example displays how many times the foreach loop repeated.

For each loop



# Vinay Tech House

## **SCRIPTING IN SSIS (CONTROL FLOW & DATA FLOW LEVEL)**

Scripting is helpful for 2 reasons  
 a) To Read (or) write values to the variables from control flow.  
 b) To implement custom operations in control (or) data flow. (If the expected functionality you don't find in SSIS tasks (or) transforms we go for custom functionality).

Scripting languages: a) VB.NET b) C-sharp

Programming protocols: 1. Similar to .NET only, supports break points, locals, watch etc debugging options.

2. follows programming hierarchy protocols such as namespace, class, method etc.

3. It supports %p variables (read) and inout variables (read, write) from package.

There is an entry point in the scripting called main, this method can be changeable.

Scripting levels: In SSIS 2 Levels. a) Control flow level b) Data flow level.

a) Script task

b) Script component

COMPONENT	WHEN TO USE
Script Task	This task is used in the Control Flow. Use this task when you need to program logic that either controls package execution or performs a task of retrieving or setting variables within a package during runtime.
Script Component	This component is used in the Data Flow. Use this component when moving data using the Data Flow Task. Here you can apply programmatic logic to massage, create, or consume data in the pipeline.

### SCRIPT TASK (.NET)

The Script Task enables you to access the Microsoft Visual Studio Tools for Applications (VSTA) environment to write and execute scripts using the VB and C# languages. Using this task, you can create additional logic that the canned SSIS tasks can't accomplish. The latest addition to SSIS of the VSTA environment and the Script Task in general also offer these extra functional advantages:

- A coding environment with the advantage of IntelliSense
- An integrated Visual Studio design environment within SSIS
- An easy-to-use methodology for passing parameters into the script
- The capability to add breakpoints to your code for testing and debugging purposes
- The automatic compiling of your script into binary format for increased speed

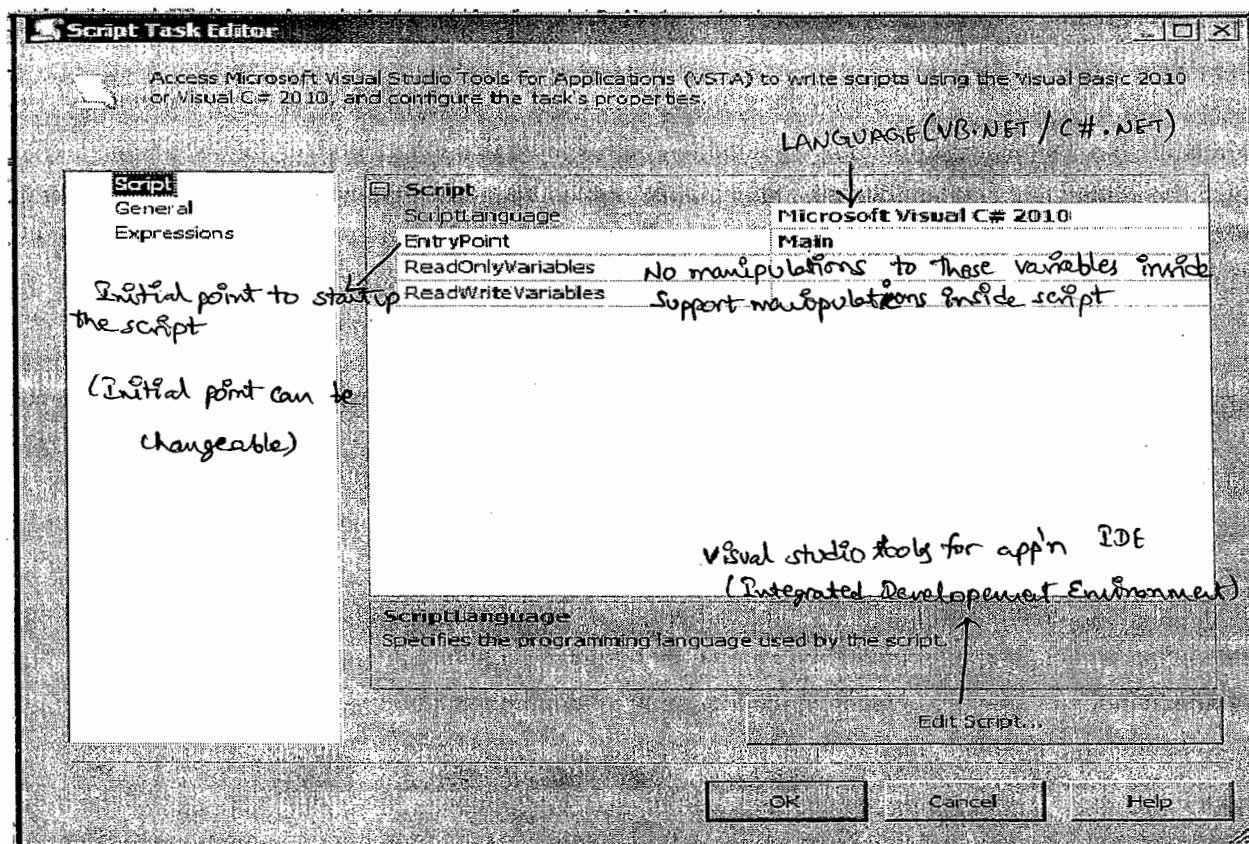
The EntryPoint property enables you to provide an alternative function to call initially when the ScriptMain class is instantiated. The ReadOnlyVariables and ReadWriteVariables properties enable you to pass SSIS variables into the script as a listing of variable names separated by commas.

When you click the actionable Edit Script button, the Visual Studio Tools for Applications environment opens to allow coding directly in the class ScriptMain. In this IDE, you have access to all the advanced debugging tactics, breakpoints, and IntelliSense found in the Visual Studio environment.

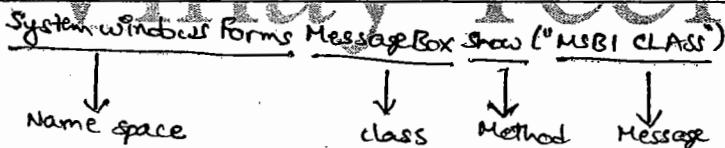
### SCRIPT COMPONENT

The Script Component enables you to write custom .NET scripts as transformations, sources, or destinations. Once you drag the component over, it will ask you if you want it to be a source, transformation, or destination. Some of the things you can do with this transformation include the following: Create a custom transformation that would use a .NET assembly to validate credit card numbers or mailing addresses. Validate data and skip records that don't seem reasonable. For example, you can use it in a human resource recruitment system to pull out candidates that don't match the salary requirement at a job code level.

Read from a proprietary system for which no standard provider exists. Write a custom component to integrate with a third-party vendor. Scripts used as sources can support multiple outputs, and you have the option of precompiling the scripts for runtime efficiency.



### SCRIPT TASK EXAMPLES (total 5)



Ex:1: Display a small text in message box using VB.NET and C-sharp.NET.

**C#**

```

using System.Windows.Forms;
...
MessageBox.Show("Hello World!");
or
System.Windows.Forms.MessageBox.Show("Hello world!");

```

**VB**

```

Imports System.Windows.Forms
...
MessageBox.Show("Hello world!")
or
System.Windows.Forms.MessageBox.Show("Hello world!");

```

Ex:2: Creating a function and calling inside main

```

/// <summary>
/// This method is called when this script task executes in the control flow.
/// Before returning from this method, set the value of Dts.TaskResult to
/// To open Help, press F1.
/// </summary>

```

```

public void Main()

```

```

    System.Windows.Forms.MessageBox.Show(GetFileName("bank file", "txt"));

```

**Vinay** # 505, Annapurna Block, Aditya Enclave, Ameerpet, Hyd. Ph: 04066638869/9573168449

```

    Dts.TaskResult = (int)ScriptResults.Success;
}

```

```
public string GetFileName(string Prefix, string Extension)
{
    return Prefix + "-" + Guid.NewGuid().ToString() + "." + Extension;
}

```

script results declaration

Example3 (GOOD EXAMPLE FOR ---- EXPORTING DATA IN XML FORMAT.

- 3) ↳
    - 1) Create XMLResult variable with String Data type in Package level
    - 2) In control flow Task Drag and Drop Execute SQL Task And Script Task
    - 3) Edit Execute SQL Task General - □ Result Set: XML

*Connection: (Connect to your Required DataBase)*

### SQL Statement:

```
SELECT * FROM dbo.EMP FOR XML AUTO,ELEMENTS,ROOT('EMP')
```

- Result Set: Result Name - 0 , Variable Name -- User::XMLresult*

#### 4) Edit Script Task

*Click Edit Script:*

*Write below script:*

## *Imports System*

*Imports System.Data  
Imports System.Math*

## *Imports System.Math*

## *Imports Microsoft.S ystem.Windows.T*

## *Imports System.Tex*

## Imports System.IO Public Sub Main()

*'Add your code here'*

```
Using Out As StreamWriter = New StreamWriter("D:\EMP xml")
```

```
OutWrite(Dts.Variables("User::XMI result").Value.ToString())
```

*Out Closet*)

#### End Using

Dts TaskResult ScriptResults Success

*End Sub*

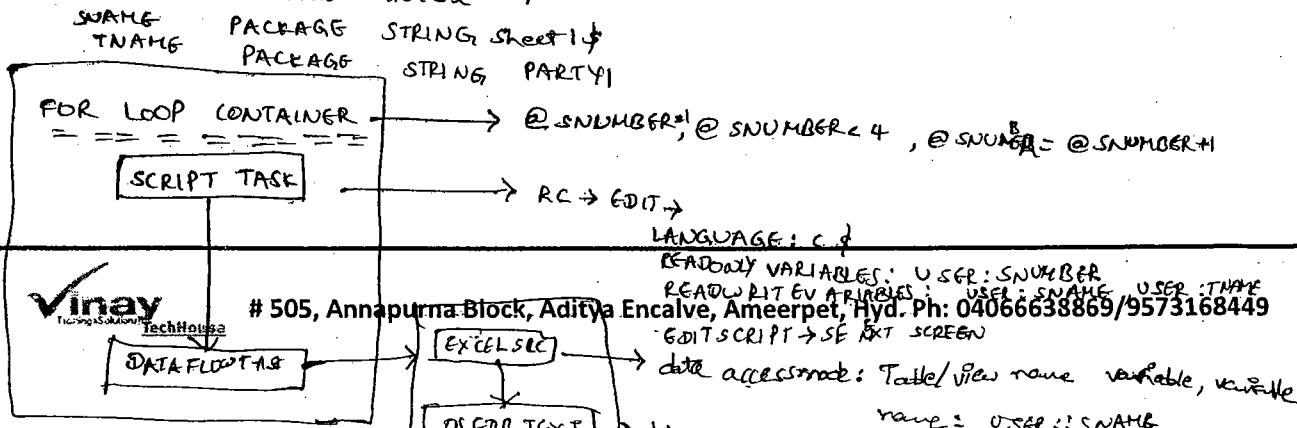
## MULTIPLE WORKSHEETS LOADING INTO MULTIPLE TABLES

There is no built-in task (or transform) available to perform this operation. So, scripting is mandatory at this situation.

CREATE THREE TABLES WITH THE BELOW STRUCTURES

```
CREATE TABLE P1 (PID NVARCHAR(20), PNM NVARCHAR(20))  
        * P2 (           " )  
        * P3 (           " )
```

- CREATE AN EXCEL SHEET WITH THREE WORK SHEETS(sheet1,sheet2,sheet3) and in each sheet take PID and PNM columns and fill with some data.
  - CREATE THREE VARIABLES



```

public void Main()
{
    // TODO: Add your code here
    Dts.Variables["SNAME"].Value = "";
    Dts.Variables["TNAME"].Value = "";
    Dts.Variables["SNAME"].Value = "Sheet" + Dts.Variables["SNUMBER"].Value + "$";
    Dts.Variables["TNAME"].Value = "PARTY" + Dts.Variables["SNUMBER"].Value;

    ...
    Dts.TaskResult = (int)ScriptResults.Success;
}

```

### MULTIPLE WORKSHEETS LOADING TO SAME TABLE

In the situation we should not change the table name dynamically in the destination because it should be fixed, for doing this in the destination specify table (xxx) view and then specify the table name.

### SCRIPTING OTHER EXAMPLES

Refer to Banking project for more examples.

### LOGGING

Logging is an important part of any data process, because it gives administrators and developers insight into what transpired during a process, with the following benefits:

- ✓ Error triage to help identify as quickly as possible what was the point and cause of the failure, such as the failure of a Lookup Component to match a record
- ✓ Root cause analysis so that a solution can be put in place to prevent a failure situation in the future
- ✓ Performance metrics such as package and execution times so that negative performance trends can be observed and addressed before the performance impact causes an ETL failure

→ SSIS contains built-in logging features that capture execution details about your packages.

→ Logging enables you to record information about events you are interested in as the package runs.

→ The logging information can be stored in a text or XML file, to a SQL Server table, to the Windows event log, or to a file suitable for Profiler.

→ Logging can be enabled for all or some tasks and containers and for all or any events.

## LOG PROVIDERS

→ Log displayable areas.

SSIS includes several default log providers. These providers are selected in the Provider type combo box and are defined as follows:

1. **SSIS Log Provider for Text Files:** This provider is used to store log information to a CSV file on the file system. It requires you to configure a File Connection object that defines the location of the file. Storing log information in a text file is the easiest way to persist a package's execution. Text files are portable, and the CSV format is a simple-to-use industry wide standard.
2. **SSIS Log Provider for SQL Server Profiler:** This provider produces a SQL Profiler trace file. The file must be specified with a trc file extension so that you can open it using the SQL Profiler diagnostic tool. Using SQL Profiler trace files is an easy way for DBAs to view log information. Using Profiler, you can view the execution of the package step by step, even replaying the steps in a test environment.
3. **SSIS Log Provider for SQL Server:** This provider sends package log events to a table in the specified SQL Server database. The database is defined using an OLE DB Connection. The first time this package is executed, a table called sysssislog is created automatically. Storing log information in a SQL Server database inherits the benefits of persisting information in a relational database system. You can easily retrieve log information for analysis across multiple package executions.
4. **SSIS Log Provider for Windows Event Log:** This provider sends log information to the Application event store. The entries created are under the Source name SQLISPackage110. No additional configuration is required for this provider. Logging package execution to the Windows event log is possibly the easiest way to store log events. The Windows event log is easy to view and can be viewed remotely if required.
5. **SSIS Log Provider for XML files:** This provider stores log information in a specified XML file on the file system. The file is specified through a File Connection object. Make sure you save the file with an xml file extension. Logging events to XML inherits the advantages of the XML specification. XML files are very portable across systems and can be validated against a schema definition.

**Example & Pic** → Taking log information outside db package at particular event and sending to mgr (or) lead for error rectification (or) performance issue this concept is helpful.

This log contains auditable information such as machine name, user name etc and technical information.

Eg: Taking log into a textfile as soon as the job execution successful

SSIS → Logging → container → <sup>PACKAGE</sup>provider → ssis log provider <sup>for text files</sup> →  ssis log provider → config → specify a filename → details → on post execute  
Execute the package and see the log information.

## CATALOG LOGGING

→ To control the log information size this is introduced in 2012

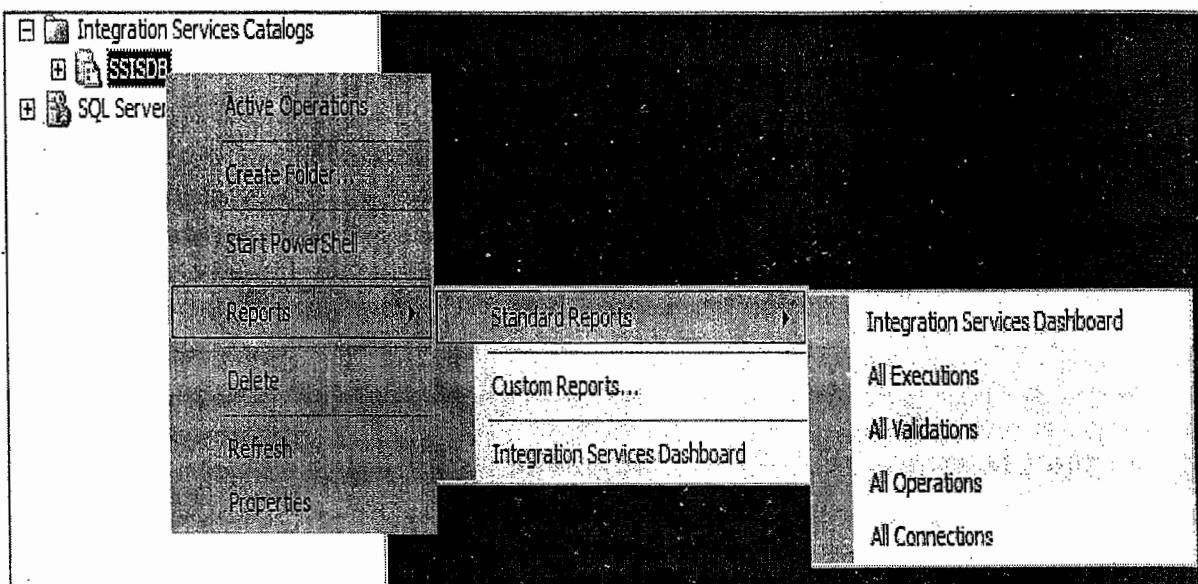
In addition to logging that can be turned on at a package level, SSIS 2012 introduces another layer of logging as part of the Integration Services catalog. If your SSIS project is set to Project Deployment Model, your deployed packages can utilize the SSIS service. This is especially handy when the package encounters a problem while running in production, rather than during development. In comparison to traditional logging, you do not need to add a logging provider to the package and can apply the catalog logging after the fact. Catalog logging also provides additional information for the administrator after the package has been deployed.

Once your project has been deployed to the server, it will automatically participate in the catalog logging. You can modify the amount of logging that occurs by changing the logging level at execution time. The logging can be set to four levels, as described in following table.

LOGGING LEVEL	DESCRIPTION
None	Includes minimal logging of executions, tasks, and parameters
Basic	Includes standard logging, as well as execution statistics and messages for specific events, including pre- and post-executions and pre- and post-validations
Performance	Includes standard logging and error and warning events and event contexts for performance tuning purposes
Verbose	Includes all prior logging categories, as well as additional performance tuning and custom task entries

# Vinay Tech House

- Under the Integration Services Catalogs node, right-click the SSISDB node. If you named your catalog differently, that name will appear here instead.
- Select Reports → Standard Reports to see the reports that are available to you, as shown in These reports enable you to see which packages have succeeded or failed or are still running. They break a package down to a more granular level and even show the task or component that could have a problem. For more detailed information on additional reports.
- If you select the All Execution report, and then click the Overview link on a package you've just run, you will see this granular detail. By running the last package you created and following these steps, you can see the successes and failures of the package, as shown in Figure 17-36. If you need to see information that is not readily available on the provided reports, you can create your own reports or write your own queries by using the SSISDB database. Figure 17-37 shows a sample of the views that can be used. By utilizing these views, you can investigate the executions of the packages. For example, the catalog.executions view provides information on the server's memory information at the start of execution.
- To create a query that will show you the longest-running objects in your package, you can use the catalog.executable\_statistics and catalog.executables views. Always be sure to filter the query on an execution identifier to limit the amount of information returned. Figure 17-38 shows the final query that returns the top five longest-running objects in the package.



Overview - 12/14/2011 57 PM - SSISMVP

**Overview**

on SSISMVP at 12/14/2011 3:57:42 PM

This report provides an overview of the package tasks and parameters, including execution or validation information.

[View Messages](#)

[View Performance](#)

**Execution Information**

Operation ID	61	Duration (sec)	58.25
Package	FromSSIS\Ch17_ProSSIS\10Logging.dtsx	Start Time	12/14/2011 3:56:44 PM
Environment		End Time	
Status	Running	Caller	SSISMVP MVP

**Execution Overview**

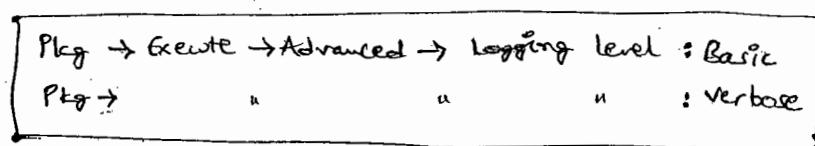
Filter: Result: All; (3 more)

Result	Duration (sec)	Package Name	Task Name	Execution Path
Succeeded	0.953	10Logging.dtsx	Empty Tables	1\0Logging\Inventory\Check Empty Tables
Failed	3.062	10Logging.dtsx	Check Inventory Level	1\0Logging\Inventory\Check Inventory Level
Failed	0.047	10Logging.dtsx	Insert Bad ProductID	1\0Logging\Inventory\Check Insert Bad ProductID

**Parameters Used**

Name	Value	Data Type
CALLER_INFO		String
DUMP_EVENT_CODE	0	String
DUMP_ON_ERROR	False	Boolean
DUMP_ON_EVENT	False	Boolean
LOGGING_LEVEL	1	Int32
SYNCHRONIZED	False	Boolean

Catalog Logging Execute the pkg in project deployment model with the below logging levels.



Pkg → Reports → All executions →

Click View Messages for the above two executions and see the log output.  
 (Verbose mode provides detailed information, Basic mode has high level information)

**PACKAGE TRANSACTION**

This part of the chapter describes how you can use transactions within your packages to handle data consistency. Two types of transactions are available in an SSIS package:

Distributed Transaction Coordinated Transaction: One or more transactions that require a DTC and can span connections, tasks, and packages

Native transaction: A transaction at a SQL Server engine level, using a single connection managed through use of T-SQL transaction commands

Note: The Microsoft Distributed Transaction Coordinator (MS DTC) allows applications to extend transactions across two or more instances of SQL Server. It also allows applications to participate in transactions managed by transaction managers that comply with the X/Open DTP XA standard. You will learn how to use them by going through four examples in detail. Each example builds on the previous example, except for the last one:

Single package: Single transaction using DTC

Single package: Multiple transactions using DTC

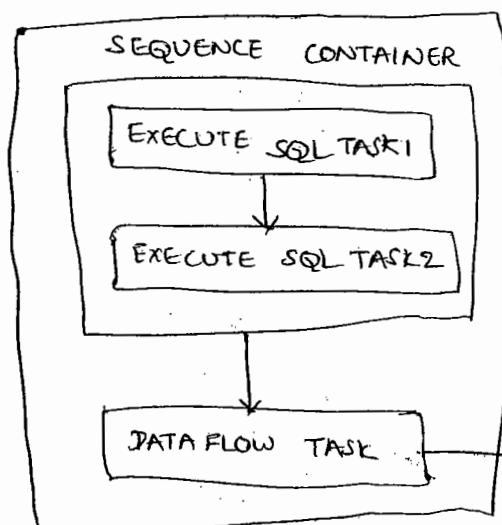
Two packages: One transaction using DTC

Single package: One transaction using a native transaction in SQL Server

For transactions to happen in a package and for tasks to join them, you need to set a few properties at both the package and the task level.

PROPERTY	VALUE	DESCRIPTION
Supported		If a transaction already exists at the parent, the container will join the transaction.
Not Supported		The container will not join a transaction, if one is present.
Required		The container will start a transaction if the parent has not; otherwise, it will join the parent transaction.

Picture:

**PACKAGE**

**Practicals:**

1. START THE SERVICE -> DTC
2. TAKE SEQUENCE CONTAINER AND KEEP TWO EXECUTE SQL TASKS.  
EXECUTE SQL TASK1->  
RC->EDIT->SQL STATEMENT: INSERT INTO EMP VALUES(9,'VINAY',50);  
RC->EDIT BREAK POINTS-> CHECK BREAK WHEN THE CONTAINER RECEIVES ON POST EXECUTE EVENT.
3. EXECUTE SQL TASK1->  
RC->EDIT->SQL STATEMENT: INSERT INTO EMP (9,'VINAY',50); (WRONG QUERY)
4. PACKAGE->CONTROL FLOW ->RC->TRANSACTION OPTION: REQUIRED  
SEQUENCE CONTAINER->RC->PROPERTIES->TRANSACTION OPTION: SUPPORTED
5. EXECUTE THE PACKAGE->AFTER THE FIRST TASK BREAKPOINT  
SSMS->DATABASE ENGINE->DB\_MSBI->RC->NEW QUERY->  
SELECT \* FROM PARTY  
SEE THE QUERY RESULT (IT DISPLAYS RECENT DATA)
6. PACKAGE->DEBUG MENU->CONTINUE  
(2ND TASK FAILED, SO SEQUENCE CONTAINER FAILS, ROLLBACK THE PREVIOUS OPERATIONS)
7. SSMS->DATABASE ENGINE->DB\_MSBI->RC->NEW QUERY->  
SELECT \* FROM PARTY;  
SEE THE QUERY RESULT (IT DISPLAYS OLD DATA SINCE TRANSACTION FAIL)

# Vinay Tech House

NOTE: MSDTC (MICROSOFT DISTRIBUTED TRANSACTION COORDINATOR)

## PRECEDENCE CONSTRAINTS

Previous component condition is called precedence constraints.

Generally to control or manage the workflow of operations this concept is helpful.  
There are 4 ways. a) Constraint b) Expression c) Expression or constraint  
d) Expression and constraint

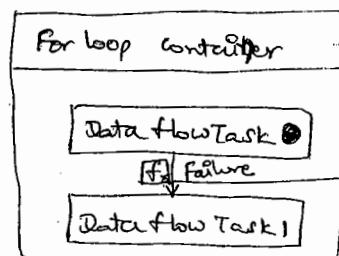
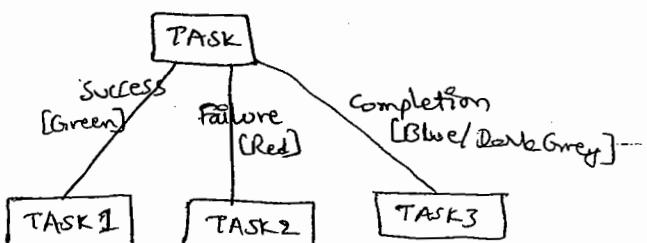
EVALUATION OPERATION	DEFINITION
a) Constraint	The execution result is applied to the constraint (success, failure, or completion) without the use of an expression.
b) Expression	Any expression that evaluates to True or False is used to evaluate the constraint without the consideration of the execution result.
c) Expression and Constraint	Both the specified execution result and an expression condition must be satisfied for the constraint to allow the next task to run.
d) Expression or Constraint	Either the specified execution result or an expression condition must be satisfied for the constraint to allow the next task to run.

Eg: Execute task1 if task is successful.

Execute task2 if task fails.

Execute task3 if task completed with success (or) failed state.

c) Expression/Condition



@i=1  
@i<=10  
@i=@i+1

Constraint: Failure  
Expression:@i<=10

A) Expression: @i=4  
B) Expression and constraint:  
Constraint: Failure  
Expression: @i=4

Expression Ex: Always expressions we take if the object is inside the loop (for, foreach)

Scenario: Execute task1 : a) If task0 executes 4 times

b) Execute task1 if task executes 4 times and failed at 4<sup>th</sup> iteration.

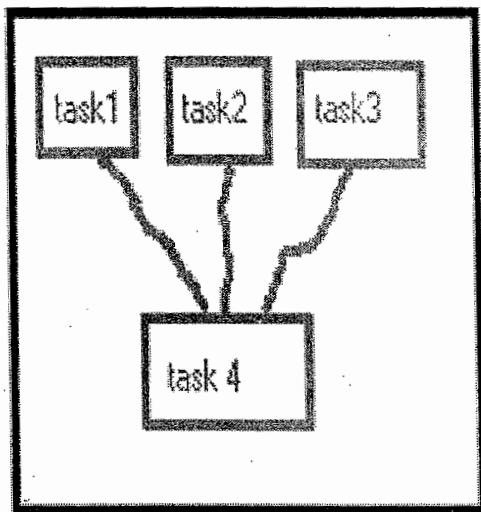
c) Execute task1 if task executes 4 times (or) failed at 4<sup>th</sup> iteration

**Multiple Constraints:** How do we execute the following task based on the preceding multiple tasks.

There are two kinds a) Logical And b) Logical OR

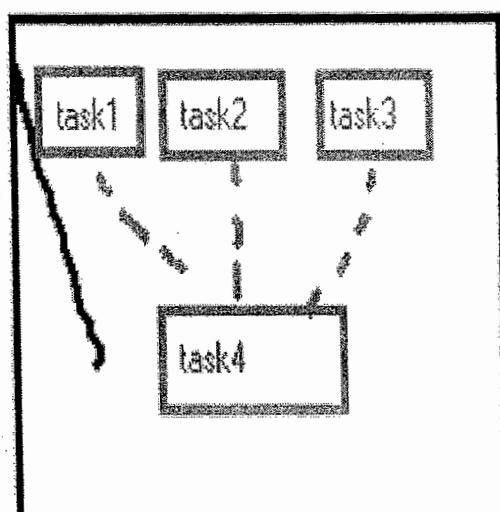
a) Logical And - All conditions must evaluate to true

b) Logical Or - One condition must evaluate to true



Logical And

THICK LINES



Logical OR

THIN LINES

## CHECKPOINT IMPLEMENTATION

It's a generic IT mechanism for implementing restartability.

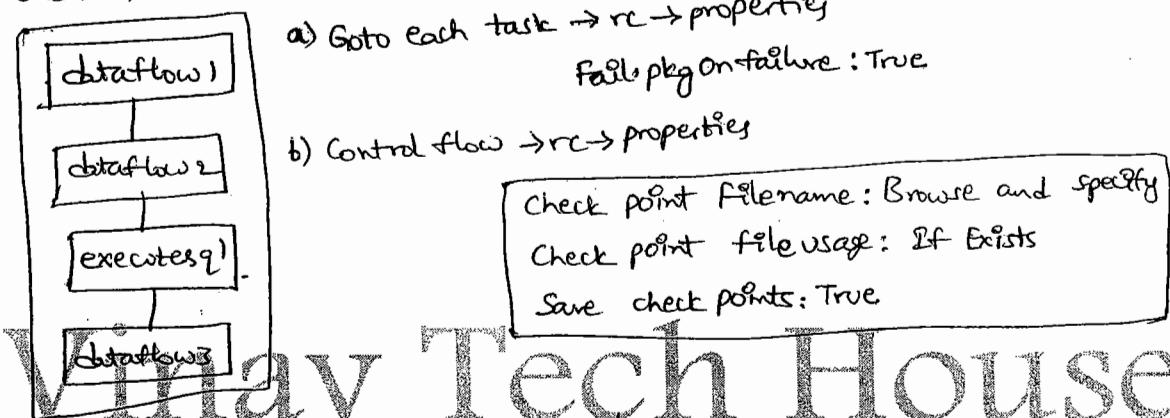
-menting restartability.

There are 2 types of restartabilities. a) Data restartability b) Task restartability

NOTE: In SSIS we have only task restartability at control-flow level data restartability at Dataflow level.

PROCESS: When app'n fails the failed point store under in a check point file, after rectification if a restart it starts from last point specified in the check point file.

- Take collection of tasks like below



# Vinay Tech House

- Take wrong strg in execute Sql task and run.

- Package fails and check point file created.

- Rectify the mistake in execute sql task and re-run the pkg so that it starts from last failed task in the pkg.

NOTE: If the pkg executed successfully check pt file automatically dropped.

Check pt file usage other options:

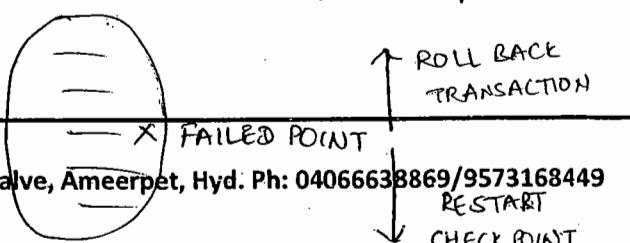
Never:

Never uses checkpoint file. Default option.

Always:

Always looks (wants) for checkpoint file.

NOTE: We can not take transactions in checkpoint in a single package.



1. To provide user interactions

2. To move apps from one environment to another environment configurations helpful in all  
**MSBI 2012 (SQL Server Integration Services)** IT applications. 95

## PACKAGE CONFIGURATIONS

1. Configurations provide user interaction to the package.
2. Generally useful at the time of working with multiple environments.  
(Dev, Testing, Production etc.)
3. Generally we store configurations in the below locations
  - a) XML file
  - b) SQL Server Database
  - c) Parent Package Variable
  - d) Environmental variable
  - e) Registry Variable

**Real time Usage:** We use most of the times XML, SQL Server configurations and some times parent package variable concept.

**\* XML Configuration:** a) Industry recommended, Universal valid, acceptable file

Generally we consider the below items or configurations.

- a) Username, password
- b) Source and destination connection strings
- c) User interaction values etc.

practices: Scenario: 1. Create pkg to the XML config and make it available by a tester to execute the pkg.

- Vinay Tech House**
- i) Create a dataflow task, and take flatfile source and destination and assign hyd ip file and hyd op file.
  - ii) Control flow → prop → configurations → click add, choose XML configurations → select source and target connections ~~seen~~ in the connection managers → finish.

Tester operation (Other resource operation): 1. Change the source and target path in the Config file.

2. START → run → DTExecUI → specify pkg path, config file path and execute.
3. Monitor the result is based on current configurations.

**\* SQL server configuration:** In the situation the config. saved in SQL table. If other people wants to work they need to perform changes in SQL table.

- i) Create an SQL server table like below,

VINAYTECH-PC\VINAY-2-dbo, SSIS-CONF		
Column name	datatype	Allow Nulls
Configuration filter	nvarchar(50)	✓
Configured value	" "	✓
Pkg path	" "	✓

In the previous pkg goto control flow → prop → click add enable config → click add → prop

**Vinay**

TechHouse # 505, Annapurna Block, Aditya Enclave, Ameerpet, Hyd. Ph: 04066638869/9573168449

Config type SQL server

Specify ...

Connection config table VINAYTECH-PC\VINAY-2-dbo.[SSIS-CONF]

click next → specify source & target → connection strings → finish.

So the pkg runs with latest settings.

## MSBI 2012 (SQL Server Integration Services)

96

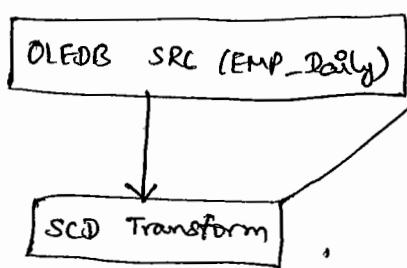
Refer to the separate hand copy material for in detail Theory explanation.

### SCD (SLOWLY CHANGING DIMENSION) PRACTICAL

SSIS Inbuilt transform supports the below methods.

- a) Fixed attribute → Here the attribute value is fixed, There is a change in the attribute value we may ignore the value change (or) fail the component.
- b) Change attribute → SCD type 1
- c) Historical attribute → SCD type 2 status and date methods.

PRACTICES:



→ TC → edit → next → specify EMP\_HIST table as SCD table → specify EID as Business key → next.

#### a) Fixed Attribute

Column	Type
ELOC	Fixed Attribute

Fail Transform → next → next → Finish

#### b) Changing Attribute (SCD Type 1)

Column	Type
ELOC	Changing Attribute

#### c) Status Mechanism (SCD Type 2)

Column	Type
ELOC	Historical Attribute

→ Next → Status column to indicate : stat value when current : current value when expired : expired

#### d) Date Mechanism (SCD Type 2)

Column	Type
ELOC	Historical Attribute

→ Next → start date column : ST\_DT  
end " " : End\_DT  
Value for dates : Container start &

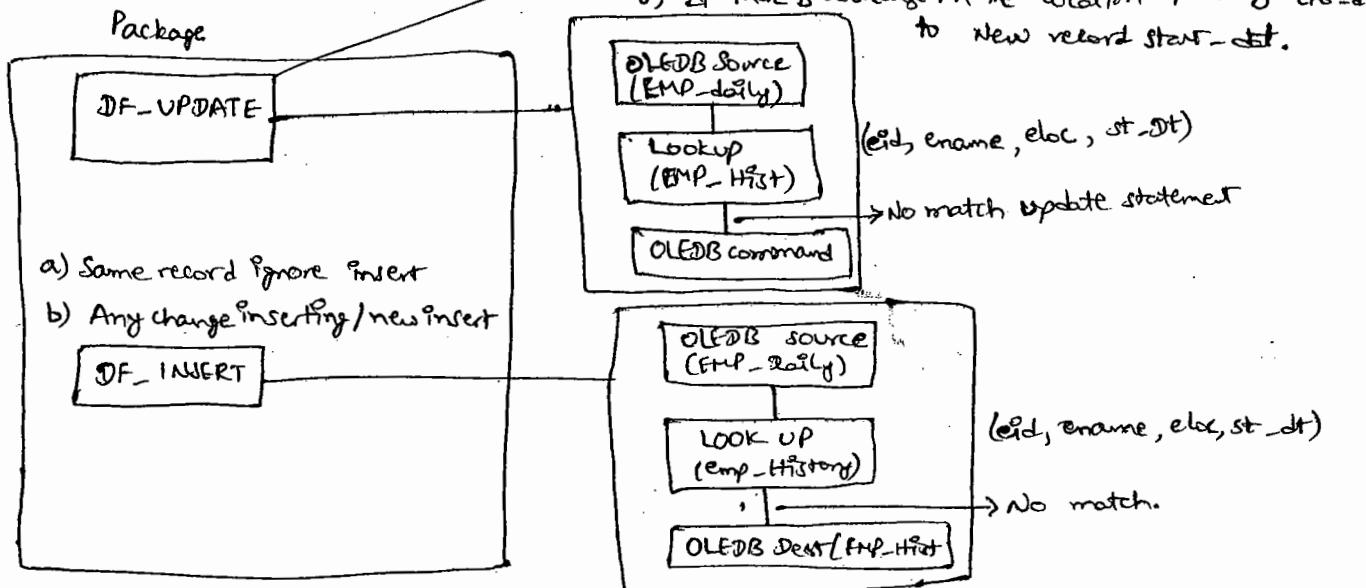
# Vinay Tech House

Specified start date follow either of the approaches.  
 a) In the SCD Transform changing the date in the flow  
**MSBI 2012 (SQL Server Integration Services)**

97

### **SCD MANUAL APPROACH (TYPE-2 DATE MECHANISM)**

b) follow the below mechanism.



#### **DF\_UPDATE:**

OLEDBSOURCE:EMP\_DAILY  
 LOOKUP:  
 RC->EDIT

#### GENERAL:

CACHEMODE:FULLCACHE

SPECIFY HOW TO HANDLE ROWS WITH NO MATCHING ENTRIES:

REDIRECT ROWS TO NO MATCH OUTPUT

#### CONNECTION:

OLEDB CONNECTION MANAGER:LOCALHOST:DB\_MSBI

USE A TABLE OR A VIEW: EMP\_HIST

#### COLUMNS:

CONNECT EID,ENAME,ELOC TO HIST TABLE FOR VALIDATION

CLICK OK,OK

#### OLEDBCOMMAND:

RC->EDIT

#### CONNECTION MANAGER TAB:

CONNECTION MANAGER:LOCALHOST:DB\_MSBI

#### COMPONENT PROPERTIES TAB:

SQLCOMMAND:UPDATE EMP\_HIST SET END\_DT=? WHERE EID=? AND END\_DT IS NULL

#### COLUMN NAMES TAB:

CONNECT EID TO PARM\_1,END\_DT TO PARM\_0

OK OK

**DF\_INSERT:**

OLEDBSOURCE:EMP\_DAILY

LOOKUP:

RC-&gt;EDIT

GENERAL:

CACHEMODE:FULLCACHE

SPECIFY HOW TO HANDLE ROWS WITH NO MATCHING ENTRIES:

REDIRECT ROWS TO NO MATCH OUTPUT

CONNECTION:

OLEDB CONNECTION MANAGER: LOCALHOST:DB\_MSBI

USE A TABLE OR A VIEW: EMP\_HIST

COLUMNS:

CONNECT EID,ENAME,ELOC TO HIST TABLE FOR VALIDATION

CLICK OK,OK

TAKE OLEDB DESTINATION AS EMP\_HIST AND CONNECT THE FILES EXCEPT END\_DT

**INCREMENTAL LOADING**

→ It is not a process, concept.

→ Incrementing destination daily data is

Called Incremental Loading.

This can also be termed as Delta loading. We implement this in various ways. Maximum database table supports incremental load.



100 records (Yesterday) →

50 more records came today →

loaded

Increment the dest

a) By direct increment ( $100 + 50 = 150$ )

b) Indirect increment (by applying some logic)

c) Using SCD process (slowly changing dimension process)

d) By using CDC (Change Data Capturing) process

Scenario: Create a pkg which loads daily current data from sql server table to a destination?

**emp-daily** → Oledb source → Data Access Mode: SQL command

Query: select \* from emp-daily where substring(convert(varchar(30), GETDATE(), 1, 11), " " " (Getdate() as char(30)), 1, 11)

Create a pkg which loads daily the previous day data from sql server table to a destination

**emp-daily**

Similar to above process with only one change replace Getdate with

Getdate() - 1

level.

Deployment is a process of moving the code from one environment to other environments  
**MSBI 2012 (SQL Server Integration Services)** 99

## DEPLOYMENT MODELS (PACKAGE & PROJECT -below 2008,2012)

In front end app's code hiding and front end display is called deployment.

Deployments and security.

### A) PACKAGE DEPLOYMENT MODEL AND PACKAGE LEVEL SECURITY

- a) Available from MSBI 2005
- b) Pkg by Pkg we deploy
- c) Manifest file required to deploy
- d) Deployment file is either file system/Sql server
- e) Pkg Level security
  - i) file system level → password protection and encryption.
  - ii) Sql server database level → Role based security.

### B) PROJECT DEPLOYMENT MODEL AND PROJECT LEVEL SECURITY

- a) Introduced in 2012
- b) Project by project deployment
- c) Requires catalog database configuration
- d) Deployment to catalog database
- e) Project Level security
  - i) file system level → project password and encryption
  - ii) Sql server catalog database level → Role based security.

# Vinay Tech House

### PROJECT DEPLOYMENT MODEL

Project is deployed.

Project uses parameters.

Project is located in an .ispac file and packages have .dtsx extensions.

Project is deployed to Integration Services Catalog.

CLR integration is required.

New environments in the SSIS catalog can be used with parameters.

Projects and packages can be validated before execution with T-SQL or managed code.

Packages are executed with T-SQL.  
Parameters and environments can be set with T-SQL.

Robust logging is built in with several reports.

### PACKAGE DEPLOYMENT MODEL

Package is deployed.

Package uses configuration files/tables.

Packages have a .dtsx extension.

Packages are deployed to the MSDB or file system.

CLR integration is not required.

System environment variables can be used with configurations.

Packages are validated just before execution and can be validated with dtexec or managed code.

Packages are executed with dtexe and dtexecui.  
Command parameters can be passed to the command prompt.

Logging is built in with no reports.

## DEPLOYMENT MODELS (PRACTICALS)

PACKAGE deployment: It requires manifest file for its operations.

Manifest file: This is the file which contains the items to be delivered.

i) Creating manifest file: 1. rc project → In solution explorer → Convert to pkg deployment model.

2. Project → properties → Deployment.

Package deployment

a) Legacy process (SQL 2008 R2)

Create deployment utility : True

b) Job Implement

Deployment path : bin\Deployment

File system Deployment: Deployment to a folder in file system

Goto manifest file in bin folder → rc → SQL server 2008 IS Pkg execution utility → Next → File system Deployment → Next → specify folder to deploy → Finish

Goto the folder → rc any pkg → Open with execute pkg utility → Execute.

SQL server deployment: Deployment to integration services service in SSMS.

Goto SSMS → IS service → stored pkgs → MUDB → rc → New folder → Decpkgs

Goto manifest file → rc execution utility → SQL server deployment → Specify folder Decpkgs → Finish.

Goto Decpkgs → Pkgname → rc → run pkg.

Required pkg's deployment:

1st way: Remove the pkg entries from Manifest file and deploy.

2nd way: Remove pkgs from deployment folder and deploy.

3rd way: Exclude pkgs from project, build solution and deploy updated manifest file.

Create a catalog database

SSMS → IS catalog → rc → Create new catalog → specify password to encrypt the content → OK

Now SSISDB created

Creating folder inside Catalog: SSISDB → Create folder → Jan\_Projects.

Deploying project in project deployment model:

Solution explorer → project → rc → Deploy → Next → specify servername and select folder (Jan-projects) to deploy → next → finish.

Observation: SSMS → SSISDB → Jan\_Projects → Project deployed with Packages.

Executing pkg: Re → Execute → OK

Display overview of the executed package

Viewing all executions: Pkg → Reports → All executions (display all the execution of the pkg)

Validity pkg: Pkg → validate

Parameterized execution: Pkg → Execute → Pass parameters (if it takes)

Version maintenance of packages: SSISDB → Properties

Number of versions: 10

Number of days: 365

Monitoring and Maintaining Versions:

a) Deploy a project from SSDT to SSISDB

b) Goto SSISDB → FOLDER name → Project → re → Version → Observe the version

c) Goto SSDT do some changes to some pkg's and deploy.

d) Goto SSISDB → Folder name → Project → re → Version → Observe now two versions <sup>1st & 2nd version</sup>  
- Which ever version pkg's you want to execute, restore them and execute them with the option provided.

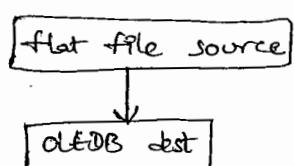
## PARAMETERS, ENVIRONMENTS (2012 introduced) - important

Vinay Tech House

1. Create two parameters like below in the pkg.

Name	Type	Value	Copy from connection string properties or connection mgr.
src_file_path	string	copy from connection string properties or connection mgr.	" " " "
dest_table_path	string	"	" " "

2. Create dataflow task like below.



- Specify source some file name and dest some db table
- Goto Source connection mgr → Expression → click ellipsis → choose connection string → click ellipsis → drag and drop src\_file\_path parameter to the expression area.
- Goto destination connection mgr → and follow the above steps. dest\_table\_path parameter

3. Deploy in project deployment model.

4. Goto SSMS → Catalog DB → Project → Pkg → re → Execute → It will prompt you source, destination parameters, pass some other file with same structure and dest as so other databases with same structure table.

→ Environments: Environments help in reducing maintenance of multiple copies of the setup. (It uses same copy pasted or same copy available in project Catalog database)

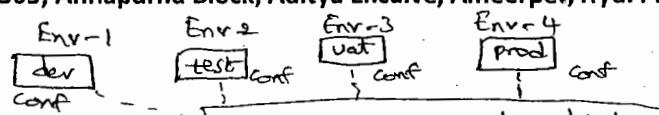
No environment:  dev  test  uat  prod

pkgs, conf pkgs, conf pkgs, conf pkgs, conf

# 505, Annapurna Block, Aditya Encalve, Ameerpet, Hyd. Ph: 04066638869/9573168449

Vinay  
TechHouse

With environment:



1. Create a pkg with parameters like belows above (SRC-FILE-PATH, DEST-FILE-PATH) and deploy to catalog database.
2. Goto SSMS → Catalog DB → Projects → Environment and right click, create two environment like below.,
- Environments → rc → name : DEV\_Env → OK  
 DEV\_Env → properties → variables → create the below two  
 Src\_File\_Path String value(connection string path)  
 Dest\_Table\_Path String value(connection string path)  
NOTE: Take development settings (Hyd file, x database)
- Environments → rc → name : Prod\_Env → OK  
 Prod\_Env → Properties → Variables → Create the below two  
 Src\_File\_Path : string value(connection string file path - actual)  
 Dest\_file\_Path " " " " " table " "  
NOTE: (Hm file, if database)
3. Goto project → configure → References → click add and add those two environments
4. Goto Project → Configure → Parameters → assign like below.,  
 pkg → Src\_File\_Path → click ellipse → specify environment variable Src\_file\_Path  
 pkg → Dest\_File\_Path → click ellipse → specify Dest\_file\_Path variable.
5. Pkg → rc → execute choose environment → OK  
NOTE: Pkg executed according to the environment settings.

## SECURITY, TYPES AND IMPLEMENTATION

**Security:** - We apply two levels at two different situations  
 a) BIDS / SSDT Level (During the development) b) SSMS Level (Role Base security)

### BIDS Level: -

Here we can use passwords and protection levels for the package level.

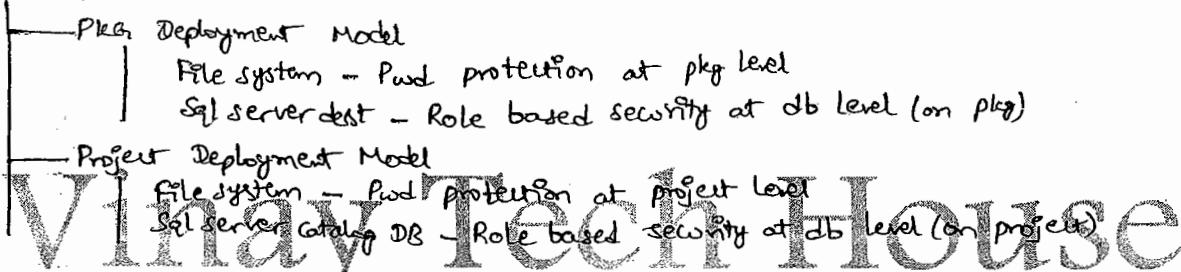
It protects both sensitive & non-sensitive information.

**Sensitive Information:** - Generally connection strings user variable etc... treated as sensitive information.

**Protection Level:** - The below protection levels are supported.

Protection level	Description
Do not save sensitive (DontSaveSensitive)	Suppresses the values of sensitive properties in the package when the package is saved. This protection level does not encrypt, but instead it prevents properties that are marked sensitive from being saved with the package and therefore makes the sensitive data unavailable to other users. If a different user opens the package, the sensitive information is replaced with blanks and the user must provide the sensitive information.
Encrypt all with password (EncryptAllWithPassword)	Uses a password to encrypt the whole package. The package is encrypted by using a password that the user supplies when the package is created or exported. To open the package in SSIS Designer or run the package by using the dtexec command prompt utility, the user must provide the package password. Without the password the user cannot access or run the package.

Encrypt all with user key (EncryptAllWithUserKey)	Uses a key that is based on the current user profile to encrypt the whole package. Only the user who created or exported the package can open the package in SSIS Designer or run the package by using the <b>dtsexec</b> command prompt utility.
Encrypt sensitive with password (EncryptSensitiveWithPa...)	Uses a password to encrypt only the values of sensitive properties in the package. DPAPI is used for this encryption. Sensitive data is saved as a part of the package, but that data is encrypted by using a password that the current user supplies when the package is created or exported. To open the package in SSIS Designer, the user must provide the package.
Encrypt sensitive with user key (EncryptSensitiveWithUs...)	Uses a key that is based on the current user profile to encrypt only the values of sensitive properties in the package. Only the same user who uses the same profile can load the package. If a different user opens the package, the sensitive information is replaced with blanks and the current user must provide new values for the.
Rely on server storage for encryption (ServerStorage)	Protects the whole package using SQL Server database roles. This option is supported only when a package is saved to the SQL Server msdb database. It is not supported when a package is saved to the file system from Business Intelligence Development Studio.

SECURITY

File system level : a) PKG

rc → properties → specify prod and protection mechanism

b) Project

Solution explorer → rc → properties → protection level and password.

NOTE: Save and close. Try to open solution then it asks passwords.

ROLL BASE SECURITY

Creating USER/ GROUP: My computer → rc → Manage → Local users/groups → Users → rc → New user → N → TEST - USER → OK

Creating Login: SSMS → Security → Logins → New login → TEST - USER → OK

Creating DB user and assigning roles and schemas:

System DB's → Master → Security → Users → rc → new user → specify windows user as

Login created → assign default roles and schemas → OK

Role based security at Legacy systems (Old versions and pkg deployment):

SSMS → IS → MSDB → PRGNAME → RC → PRG\_ROLES → Assign your role to READER/WRITER SECTION

Role based security at legacy systems (Latest versions and pkg deployment):

SSMS → IS Catalog → Project → rc → Properties → Permissions → Browse → Select roles → Specify grant and deny options on the objects to the roles.

## COMMAND LINE UTILITIES

→ To execute pkg's other than SSIS or SQL server these are helpful.

DTExec is a tool you use to execute the pkg's from a command line (start → run) and DTUtil can help you migrate a pkg or change the security of a pkg, just to name a couple of its functions. In 2012 command line tools have been enhanced to support executing pkg's in the SSIS catalog.

### DTExec

DTExec is a command-prompt tool included with SQL Server. This command is used to configure and execute SSIS packages. It gives access to all the package configuration and execution options, like connections, properties, variables, logging, and progress indicators. It also supports packages from three different sources: **SQL Server databases, the SSIS package store, and the file system.**

DTExecUI is a powerful tool that wraps the command-line utility DTExec. A shortcut here is to use DTExecUI to create the command for you. You can see the list of switches, minus three optional switches, for this utility by typing the following:

**dtexec.exe /?**

For example, to execute a package that is stored in the MSDB database on your localhost, you could use the following command. This command is more verbose than is required. In reality, you only need to type the /DTS and /SERVER switches to find and execute the package.

**NTAX:** **DTExec.exe /DTS "MSDB\DBSnapshots" /SERVER localhost /MAXCONCURRENT " -1 "**  
**/CHECKPOINT OFF /REPORTING V**

TABLE 22-3: DTEXEC Options

OPTION	DESCRIPTION
/ISServer package_path	(Optional) Loads a package that is on an SSIS server. The package_path points to the path and filename of the package.
/Par[ameter] parameter_name (Type);Value	(Optional) Sets a parameter value. This can be a project level or package level parameter.
/Envreference reference_id	(Optional) Sets the environment to be used by the package.

### DTExecUI

In older versions of SQL Server, the primary way to execute a package was with DTExecUI.exe. With the integration of T-SQL to execute the packages, these tools will not be needed as often. This utility is a graphical wrapper for DTExec.exe, and it provides an easier way to produce the necessary switches to execute the package. You can open the utility by selecting Start → Run and typing **DTExecui.exe**.

Navigation : → start → run → DTEXECUI → Specify pkg name, configuration name and click execute.

## DTUTIL

One of the best undiscovered command-line tools in your administrator kit is DTUtil.exe. This is also a good tool for developers. It performs a number of functions, including moving packages, renumbering the PackageID, re-encrypting a package, and digitally signing a package. To see everything this tool can do, type the following command from a command prompt:

### DTUtil.exe /?

Essentially, this tool can be used to do many of the things that you do in Management Studio, and to a lesser extent in SQL Server Data Tools. The next sections describe creative ways to use DTUtil.exe.

## ERROR HANDLING TYPES & IMPLEMENTATION (REAL TIME CONCEPT)

Programming level exception handling, Tool level error handling are same. Both perform action for run-time event.

This is different from error capturing and error rectification.

Process: In error handling if there is a failure system performs the specified failure action and passes the application.

There are many ways to perform this.

- By using precedence constraints at pkg control flow level.



- By using redirection operation (Redlink) at Dataflow level.

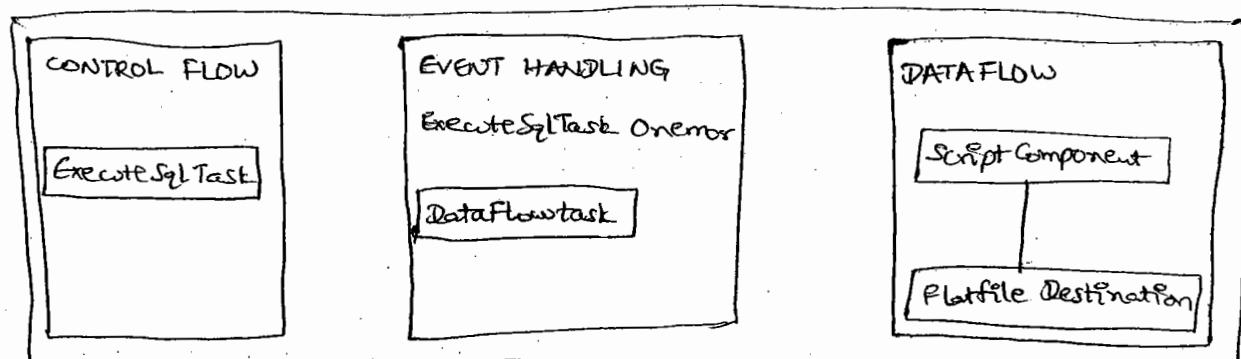


- By taking event handling operation. Event → On failure.

- By using custom error handling.

(In the below example the system error msg we are capturing in another destination and keeping the pkg in success state).

### PACKAGE



- Execute Sql Task

Connection : DB\_M8BI

Sql Statement : SELECT CONVERT(INT, "VINAYTECH") AS DWHT;

**2. Event Handler**

Event Handler      Event

ExecuteSql task      On Error

Take Dataflow task and design Dataflow like below.,

# Vinay Tech House

**3. ScriptComponent as Src and FlatFile as Destination**

**Scriptcomponent:**

**Script**

**Readonly variables: System::ErrorDescription**

**EditScript->**

```
Public Overrides Sub CreateNewOutputRows()
With OutputBuffer
    .AddRow()
    .Errdescription = Me.Variables.ErrorDescription
```

**End With**

**End Sub**

**Inputs and Outputs**

**Output**

**Outputcolumns**

**ErrDescription->DataType : DT\_STR**

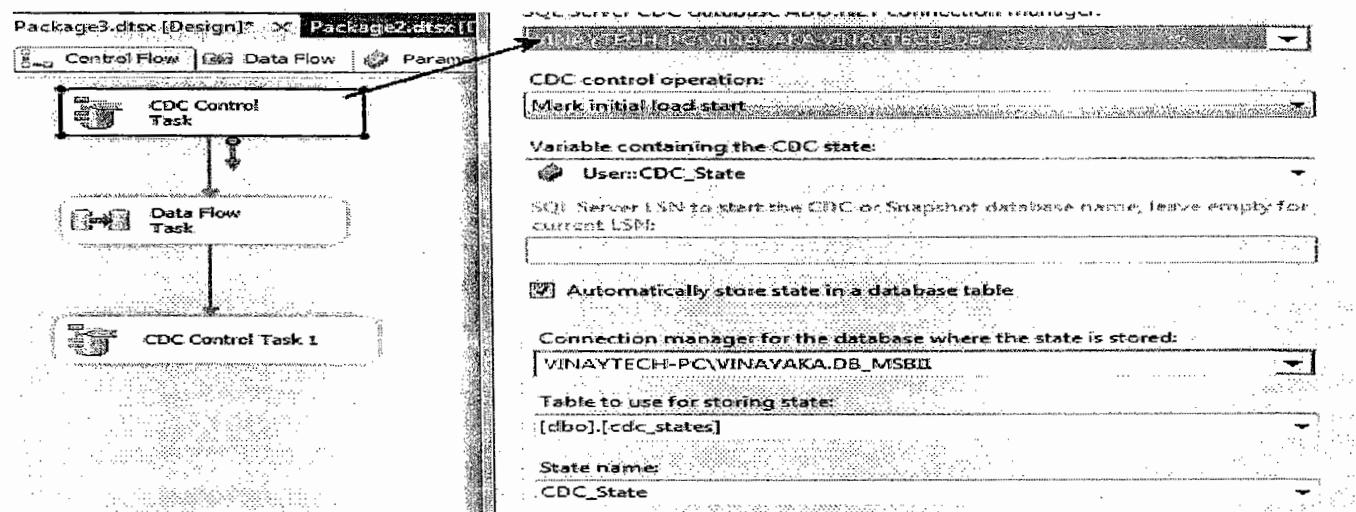
**Length : 500**

## CHANGE DATA CAPTURING (CDC), CDC TASKS

```
EXEC sys.sp_cdc_enable_db
```

ENABLING CDC AT TABLE LEVEL

```
EXEC sys.sp_cdc_enable_table
@source_schema = N'dbo',
@source_name    = N'PARTY',
@role_name      = N'VINAYAKA',
@supports_net_changes = 1
```



As a developer I never involve in scheduling. I heard like my supporting using SQL Server agent and control M for operations.

## MSBI 2012 (SQL Server Integration Services)

108

### SCHEDULERS, JOBS CREATION, ALTERING AND MANIPULATING

Scheduling: Running an application in a stipulated period regularly (daily or weekly or monthly etc).

Scheduling Tools: There are many available (native and third party).

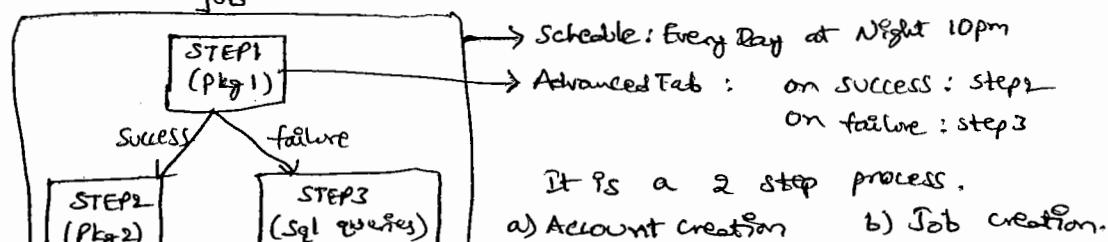
Native -- Windows Scheduler, SQL Server Agent

Third-party -- Control-M, AutoSys, Trillium, CA-7, ICL, OPUS etc...

Working with SQL Server Agent: It follows job terminology for its operations.

Job: It contains group of steps, each step may represent any application such as ~~SSIS~~ T-SQL, SSIS, AS etc.

Scenario: SQL Server Agent Jobs



A) SQL Server Agent Account creation

a) Credential creation

SSMS->DB Engine->Security->  
Credentials->RC->New credentials->  
Credential Name: VINAY\_Credential  
Identity : Administrator (Your computer connected user name and pwd)  
Password : (password of the user)  
Confirm pwd : Specify Click ok

b) Proxy Creation

SSMS->DB Engine->SQL Server Agent->  
Proxies->RC->New Proxy-->

Proxy name: vinay\_proxy  
Credential name: VINAY\_Credential  
Activity to the sub items:

SQL Server Integration Services

B) Job Creation:

1. START → Sql Server Agent

2. Jobs → New Job →

Name:   
 General x: MSBI-JOB

Steps

New Step → Name: step1

Type: IS

Run as: select previous created proxy

Pkg source: File system

Pkg Name: Browse pkg1

Advanced: On success

step2  
On failure

step3

OK

New step → Name: step2

Type: IS

Run as: select previous created proxy

Pkg source: File system

Pkg Name: Browse pkg2

New step → Name: step3

Type: IS

Run as: select previous created proxy

Pkg source: File system

Pkg name: Browse pkg3

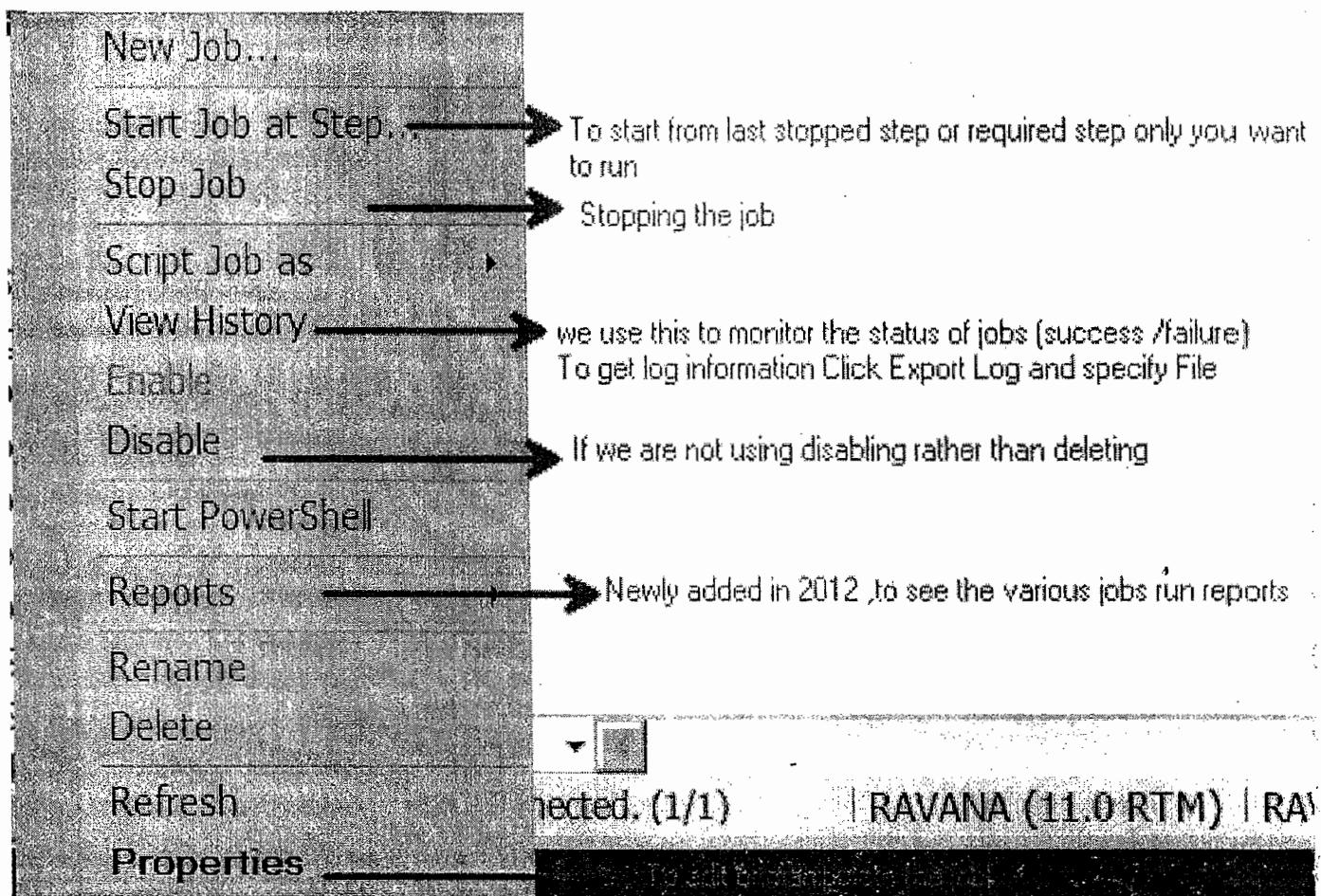
Schedule

specify recurring / one time and time to execute, start date and end date OK.

→ Why companies prefer 3<sup>rd</sup> party schedulers than native schedulers.

Sol: a) Easy to operate, navigate and full support to GUI

b) Support working with heterogeneous application.



2. What kind of tuning methods you followed in your project.

## PERFORMANCE TUNING

as performance tuning.

Identifying bottleneck: There are various options available a) Progress tab b) Log provider c) Catalog logging etc.

In the above situations we need to identify step by step process for the corresponding package.

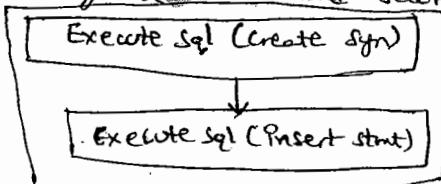
Bottle neck: It is not an error but causes system delay in performance.

Bottle neck levels: Bottlenecks could be in any levels and solution also in accord to level.

- a) Pkg level    b) Dataflow level    c) Source level    d) destination level. e) Transform level  
f) System level.

- a) Pkg level Tips:
  - 1. For complex pkg's (or) more task pkg's take checkpoints. so that you will not loose the performance.
  - 2. Disable unnecessary event handlers performance improve.
  - 3. Increase max. error count to prevent from immediate failure.
  - 4. Take a proper logging mode.
  - 5. Use transactions appropriately.
  - 6. Use delay validation property wherever required.

Eg: First task creating a table and second task inserting the data into the table.



When we run this pkg, second task fails due to table unavailability. To continue the execution happily make the validation delay of the 2nd task by taking the property DELAY VALIDATION: TRUE for the 2nd task.

- b) Dataflow level Tips:

Binary Large Object

BLOBTempStoragePath → (Images, Video files, Media files, Binary Data)  
Here system uses local drive memory for operations

Buffer TempStorage Path

Default Buffer MaxRows → Increasing buffer rows advantage 10000 when we are processing more records

Default Buffer Size → If we increase/decrease buffer rows 1024 & 5120, it is recommended to change the size.

Engine Threads → Increasing threads will also increase parallelism

Logging mode → Normal logging mode will always give good we can set performance.

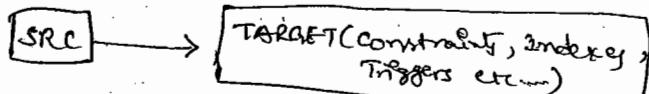
Run In Optimized Mode → Avoids unnecessary transformations and runs always in optimized fashion.

c) Source level :

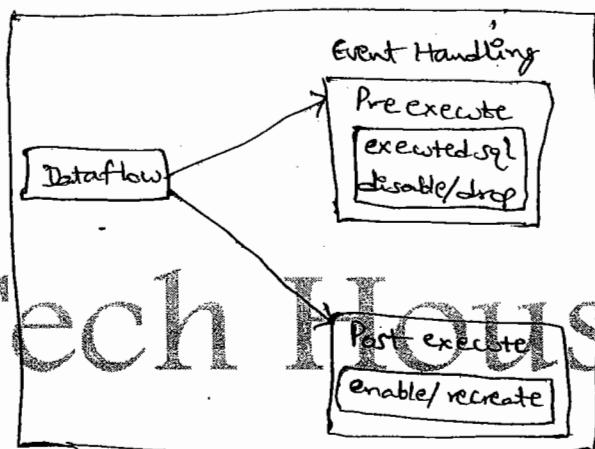
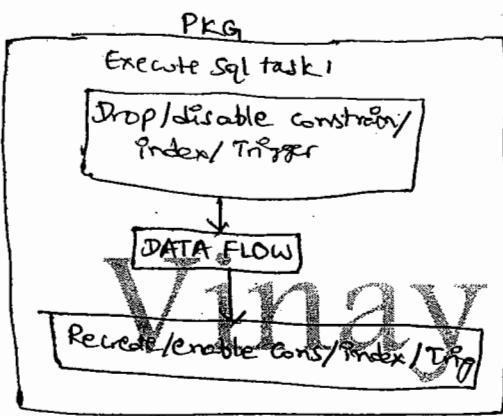
- i) In case of Flatfile: a) Take file local to the system, so it would run faster (as there is no network)
  - b) Use fastparse = True for faster and better retrieval and loading.
- ii) In case of DB: a) Use indexes for faster retrieval
  - b) Use data access mode as sql command for customized query processing.

d) destination level :

- a) File or flatfile: Take file local to the system
- b) Incase of DB: Use dataaccess mode as sql command for loading in customized way.



Two ways:

1st way2nd way

Vinay Tech House

- e) Transform level:
- i) use multicast for single source reading and to perform various operations to the source
  - ii) use OLEDB command to perform operation on every row.
  - iii) use sort Transform with duplicate row removal option to eliminate duplicates.
  - iv) use merge for sorted inputs.
  - v) use merge join to implement outer join operation.
  - vi) use lookup for exact match.
  - vii) use lookup and cache for optimal performance.
  - viii) use pivot to normalize the data.

NOTE: It is recommended to take bulk insert task and fast load options for more <sup>voluminous</sup> data.

- f) System level Tips:
- i) There are no bottlenecks in the above all levels definitely the problem should be at system level.
  - ii) DBA's and other senior resources will participate to eliminate them.

NOTE: Refer to next pages for detail and more optimization methods.

## IN DEPTH DATAFLOW (TO TUNE PERFORMANCE)

Four important components.

Data buffer architecture

Transformation types

Transformation communication

Execution trees

### **Memory Buffer Architecture**

The Data Flow manages data in groups of data called *buffers*. A buffer is merely memory that is allocated for the use of storing rows and columns of data where transformations are applied. This means that as data is being extracted from sources into the engine, it is put into these preallocated memory buffers.

Buffers are dynamically sized based on row width (the cumulative number of bytes in a row) and other package and server criteria. A buffer, for example, may include

9,000 rows of data with a few columns of data.

Figure 15-4 shows a few groupings of buffers.

The diagram illustrates three separate buffers, labeled Buffer 1, Buffer 2, and Buffer N. Each buffer contains a table with the following columns: now, num, col 1, col 2, col 3, col 4, col 5, col 6, and col 7. The data is as follows:

- Buffer 1:** Contains rows 1 through 9000. The 'now' column has values 1, 2, 3, ..., 9000. The 'num' column has values 1, 2, 3, ..., 9000. All other columns (col 1 to col 7) have values represented by dots (..).
- Buffer 2:** Contains rows 9001 through 18000. The 'now' column has values 9001, 9002, 9003, ..., 18000. The 'num' column has values 1, 2, 3, ..., 9000. All other columns (col 1 to col 7) have values represented by dots (..).
- Buffer N:** Contains rows 8000\*n+1 through 8000\*n+3. The 'now' column has values 8000\*n+1, 8000\*n+2, 8000\*n+3, ..., 8000 \* (n+3). The 'num' column has values 1, 2, 3, ..., 9000. All other columns (col 1 to col 7) have values represented by dots (..).

### **Types of Transformations**

The transformations in the Data Flow have certain characteristics that group each into different categories. The base-level differences between them are the way they communicate with each other, and how and when data is handed off from one transformation to another. Evaluating transformations on two fronts provides the background you need to understand how the buffers are managed:

**Blocking nature:** Non-blocking (sometimes called streaming), semi-blocking, blocking

**Communication mechanism:** Synchronous and asynchronous

### **Non-Blocking, Streaming, and Row-Based Transformations**

Most of the SSIS transformations are non-blocking. This means that the transformation logic applied in the transformation does not impede the data from moving on to the next transformation after the transformation logic is applied to the row. **Two categories of non-blocking transformations exist: streaming and row-based.**

The difference is whether the SSIS transformation can use internal information and processes to handle its work or whether the transformation has to call an external process to retrieve information it needs for the work.

*Streaming* transformations are usually able to apply transformation logic quickly, using precached data and processing calculations within the row being worked on. In these transformations, it is usually the case that a transformation will not slip behind the rate of the data being fed to it. These transformations focus their resources on the CPUs, which in most cases are not the bottleneck of an ETL system.

Therefore, they are classified as streaming.

The following transformations stream the data from transformation to transformation in the Data Flow:

*Audit*

*Cache Transform*

*Character Map*

*Conditional Split*

*Copy Column*

*Data Conversion*

*Derived Column*

*Lookup (with a full-cache setting)*

*Multicast*

*Percent Sampling*

*Row Count*

*Script Component* (provided the script is not configured with an asynchronous output, which is discussed in the

*"Advanced Data Flow Execution Concepts" section)*

*Union All* (the Union All acts like a streaming transformation but is actually a semi blocking transformation because it

*communicates asynchronously*)

The second grouping of non-blocking transformations is identified as *row-based*. These transformations are still non-blocking in the sense that the data can flow immediately to the next transformation after the transformation logic is applied to the buffer. The row-based description indicates that the rows flowing through the transformation are acted on one by one with a requirement to interact with an outside process such as a database, file, or component.

The following transformations are classified as row-based:

*DQS Cleansing*

*Export Column*

*Import Column*

*Lookup (with a no-cache or partial-cache setting)*

*OLE DB Command*

*Script Component* (where the script interacts with an external component)

*Slowly Changing Dimension* (each row is looked up against the dimension in the database)

### Semi-Blocking Transformations

The next category of Transformation Components are the ones that hold up records in the Data Flow for a period of time before allowing the memory buffers to be passed downstream. These are typically called semi-blocking transformations, given their nature. Only a few out-of-the-box transformations are semi-blocking in nature:

*Data Mining Query*

*Merge*

*Merge Join*

*Pivot*

*Term Lookup*

*Unpivot*

*Union All (also included in the streaming transformations list, but under the covers, the Union All is semi-blocking)*

Note:

*Semi-blocking transformations require a little more server resources than non-blocking transformations because the buffers need to stay in memory until the right data is received.*

### **Blocking Transformations**

The final category of transformation types is the actual blocking transformation. These components require a complete review of the upstream data before releasing any row downstream to the connected transformations and destinations. The list is also smaller than the list of non-blocking transformations because there are only a few transformations that require “blocking” all the data to complete an operation.

Here is the list of the blocking transformations:

*Aggregate*

*Fuzzy Grouping*

*Fuzzy Lookup*

*Row Sampling*

*Sort*

*Term Extraction*

*Script Component (when configured to receive all rows before sending any downstream)*

*The two most widely used examples of the blocking transformations are the Sort and the Aggregate; Each of these requires the entire data set before handing off the data to the next transformation.*

**Note:**

*Blocking transformations are usually more resource intensive than other types of transformations for two main reasons. First, because all the data is being held up, either the server must use a lot of memory to store the data or, if the server does not have enough memory, a process of file staging happens, which requires the I/O overhead of staging the data to disk temporarily. Second, these transformations usually put a heavy burden on the processor to perform the work of data aggregation, sorting, or fuzzy matching.*

### **Asynchronous Transformation Outputs**

It is easier to begin with the *asynchronous* definition because it leads into a comparison of the two kinds of transformation outputs, *synchronous* and asynchronous. A transformation output is asynchronous if the buffers used in the input are different from the buffers used in the output. In other words, many of the transformations cannot both perform the specified operation and preserve the buffers (the number of rows or the order of the rows), so a copy of the data must be made to accomplish the desired effect.

The Aggregate Transformation, for example, may output only a fraction of the number of rows coming into it; or when the Merge Join Transformation has to join two data sets together, the resulting number of rows may not be equivalent to the number of input rows. In both cases, the buffers are received, the processing is handled, and new buffers are created.

### **Synchronous Transformation Outputs**

A synchronous transformation is one in which the buffers are immediately handed off to the next downstream transformation at the completion of the transformation logic. If this sounds like the definition for streaming transformations, that’s because there is almost complete overlap between streaming transformations and synchronous transformations. The word *buffers* was intentionally used in the definition of synchronous outputs, because the important point is that the same buffers received by the transformation input are passed out the output.

Note: A transformation is not limited to a single synchronous output. Both the Multicast and the Conditional Split can have multiple outputs, but all the outputs are synchronous.

Synchronous transformation outputs preserve the sort order of incoming data, whereas some of the asynchronous transformations do not. The Sort, Merge, and Merge Join asynchronous components, of course, have sorted outputs because of their nature, but the Union All, for example, does not.

## Source and Destination Adapters

Source and Destination adapters are integral to the Data Flow and therefore merit brief consideration in this chapter. In fact, because of their differences in functionality, sources and destinations are therefore classified differently.

First of all, in looking at the advanced properties of a Source adapter, the source will have the same list of external columns and output columns. The external columns come directly from the source and are copied into the Data Flow buffers and subsequently assigned LineageIDs. While the external source columns do not have LineageIDs, the process is effectively the same as an asynchronous component output. Source adapters require the allocation of buffers, where the incoming data can be grouped and managed for the downstream transformations to perform work against.

Destination adapters, conversely, de-allocate the buffer data when it is loaded into the destinations. The advanced properties of the Destination adapter include an External Column list, which represents the destination columns used in the load. The input columns are mapped to this External Column list on the Mapping page of the Destination adapter editor. In the advanced properties, you should note that there is no primary Output Container (besides the Error Output) for the Destination adapter, as the buffers do not flow through the component but rather are committed to a DestinationAdapter as a final step in the Data Flow.

## Advanced Data Flow Execution Concepts

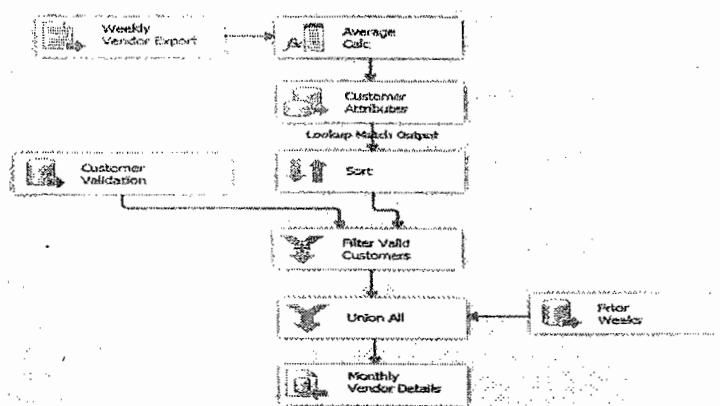
The preceding discussion of transformation types and how outputs handle buffers leads into a more advanced discussion of how SSIS coordinates and manages Data Flow processing overall. This section ties together the discussion of synchronous and asynchronous transformations to provide the bigger picture of a package's execution.

Relevant to this discussion is a more detailed understanding of buffer management within an executing package based on how the package is designed.

### EXECUTION TREES

An *execution tree* is a logical grouping of Data Flow Components (transformations and adapters) based on their synchronous relationship to one another. Groupings are delineated by asynchronous component outputs that indicate the completion of one execution tree and the start of the next.

Figure 15-12 shows the execution trees of a moderately complex Data Flow that uses multiple components with asynchronous outputs. The "paths" indicated are numbered based on the SSIS Data Flow logging, not the order in which the data flows. See the PipelineExecutionTrees log example in the "Monitoring Data Flow Execution" section of this chapter.



Recall that components with asynchronous outputs use different input buffers. The input participates in the upstream execution tree, while the asynchronous output begins the next execution tree. In light of this, the execution trees for Figure 15-12 start at the Source adapters and are then completed, and a new execution tree begins at every asynchronous transformation.

Execution trees are base 0, meaning you count them starting with a 0. In the next section, you will see how the pipeline logging identifies them. Although the execution trees seem out of order, you have used the explicit order given by the pipeline logging.

When SSIS executes a package, the buffer manager defines different buffer profiles based on the execution trees within a package. All the buffers used for a particular execution tree are identical in definition. When defining the buffer profile for each execution tree, the SSIS buffer manager looks at all the transformations used in the execution tree and includes every column in the buffer that is needed at any point within the execution tree. Note that execution tree path #5 in Figure 15-12 contains a Source adapter, a Derived Column Transformation, and a Lookup.

#### Note:

*One final note about execution trees — the process thread scheduler can assign more than one thread to a single execution tree if threads are available and the execution tree requires intense processor utilization. Each transformation can receive a single thread, so if an execution tree has only two components that participate, then the execution tree can have a maximum of two threads. In addition, each source adapter receives a separate thread.*

### Monitoring Data Flow Execution

Built into SSIS 2012 is the capability to report on Data Flow performance and identify execution trees and threads. The reports and log events can be very useful in understanding your Data Flow and how the engine is managing buffers and execution.

First of all, pipeline logging events are available in the Logging features of SSIS. As shown in the following list, several logging events provide details about what is happening in the Data Flow.

#### Note:

*When executing a package on the SSIS server, you can turn on the Verbose logging to capture all the events. This is found on the Advanced tab of the Execute Package dialog under Logging level.*

On the Details tab of this Configure SSIS Logs dialog, the two execution information log events just listed are available to capture. When the package is run, these events can be tracked to the selected log provider as defined. However, during development, it is useful to see these events when testing and designing a package. SSIS includes a way to see these events in the Business Intelligence Development

Studio as a separate window. The Log Events window can be pulled up either from the SSIS menu by selecting Log Events or through the View menu, listed under the Other Windows submenu. This window can float or be docked in the designer.

**Note: See that for your operations and continue**

## SSIS DATAFLOW DESIGN AND TUNING

Designing a data-processing solution requires more than just sending the source data into a blackbox transformation engine with outputs that push the data into the destination. Of course, system requirements will dictate the final design of the process, including but not limited to the following:

- Source and destination system impact
- Processing time windows and performance
- Destination system state consistency
- Hard and soft exception handling and restartability needs
- Environment architecture model, distributed hardware, or scaled-up servers
- Solution architecture requirements, such as flexibility of change or OEM targeted solutions
- Modular and configurable solution needs
- Manageability and administration requirements

### Data Flow Design Practices

Keep the following four design practices in mind when creating new packages:

- Limit synchronous processes.
- Monitor the memory use of blocking and semi-blocking transformations.
- Reduce staging and disk I/O.
- Reduce reliance on an RDBMS. The four architecture best practices described in the preceding section relate directly to the value that the Data Flow provides.

**Limit synchronous processes:** By bringing more of the processing logic into the Data Flow, the natural result is fewer process-oriented steps that require completion before proceeding. In the previous chapter, you looked at the general streaming nature of the Data Flow. This translates to reduced overall processing times.

**Monitor the memory use of blocking and semi-blocking transformations:** When memory becomes scarce on your server, SSIS begins to copy buffers to disk or spool them to disk. Once this happens, your package slows down dramatically. The most intensive memory transformations are blocking and semi-blocking transformations. However, other transformations, like the cached Lookup, also can require a lot of memory if the Lookup table contains millions of items. All these transformations perform very well until they near the threshold of memory on your server. It is best to monitor the memory to ensure that you avoid a low-memory situation. In particular, throttle the memory of SQL Server if it is on the same server where your SSIS packages are running.

**Reduce staging and expensive IO operations:** The Data Flow performs most operations in memory (with occasional use of temp folders and some interaction with external systems).

- Whenever processing happens on data that resides in RAM, processing is more efficient.
- Disk I/O operations rely on the performance of the drives, the throughput of the I/O channels, and the overhead of the operating system to write and read information to and from the disk. With high volumes or bursting scenarios typical with data processing and ETL, disk I/O is often a bottleneck.

**Reduce reliance on RDBMS:** Relational engines are powerful tools, and the point here is not to detract from their appropriate uses to store and manage data. By using the Data Flow to cleanse and join data rather than the RDBMS, the result is reduced impact on the relational system, which frees it

up for other functions that may be higher priority. Reading data from a database is generally less expensive than performing complex joins or complicated queries.

- In addition, related to the first bullet, all RDBMS operations are synchronous. Set-based operations, while they are very useful and optimized in a relational database system, still require that the operation be complete before the data is available for other purposes. The Data Flow, conversely, can process joins and Lookups and other cleansing steps in parallel while the data is flowing through the pipeline. However, note that an RDBMS engine can be leveraged in certain ways; for example, if a table has the right indexes, you can use an ORDER BY, which may be faster than an SSIS Sort Transformation.

## Optimizing Package Processing

### Optimizing Buffers, Execution Trees, and Engine Threads

Each execution tree in a Data Flow, a different buffer profile is used. This means that downstream execution trees may require different columns based on what is added or subtracted in the Data Flow. You also saw that the performance of a buffer within a Data Flow is directly related to the row width of the buffer. Narrow buffers can hold more rows, enabling higher throughput.

Some columns that are used in an execution tree may not be needed downstream. For example, if an input column to a Lookup Transformation is used as the key match to the reference table, this column may not be needed after the Lookup and therefore should be removed before the next execution tree. SSIS does a good job of providing warnings when columns exist in an execution tree but are not used in any downstream transformation or destination adapter. Figure 15-19 shows the Execution Results tab (also called the Progress tab when in runtime mode) within a package for which column usage has not been optimized in the Data Flow. Each warning, highlighted with a yellow exclamation point, indicates the existence of a column not used later in downstream components and which therefore should be removed from the pipeline after initial use.

For example, in the Union All Transformation, you can highlight a row in the editor and delete it with the Delete keyboard key. This ensures that the column is not used in the next execution tree.

A second optimization technique involves increasing processor utilization by adding the more execution threads for the Data Flow. As highlighted in the last chapter, increasing the `EngineThreads` Data Flow property to a value greater than the number of execution trees plus the number of Source Components ensures that SSIS has enough threads to use.

## Careful Use of Row-Based Transformations

Row-based transforms, as described earlier in this chapter, are non-blocking transformations, but they exhibit the functionality of interacting with an outside system (for example, a database or file system) on a row-by-row basis. Compared with other non-blocking transformations, these transformations are slower because of this behavior. The other type of non-blocking transformation, streaming, can use internal cache or provide calculations using other columns or variables readily available to the Data Flow, making them perform very fast. Given the nature of row-based transformations, their usage should be cautious and calculated.

Of course, some row-based transformations have critical functionality, so this caution needs to be balanced with data-processing requirements. For example, the Export and Import Column Transformation can read and write from files to columns, which is a very valuable tool, but it has the obvious overhead of the I/O activity with the file system.

Another useful row-based transformation is the OLE DB Command Transformation, which can use input column values and execute parameterized queries against a database, row by row. The interaction with the database, although it can be optimized, still requires overhead to process. Figure 15-20 shows a SQL Server Trace run against a database that is receiving updates from an OLE DB Command Transformation.

## Troubleshooting Data Flow Performance Bottlenecks

The pipeline execution reports (reviewed earlier in the chapter) are a great way to identify which component in your Data Flow is causing a bottleneck. Another way to troubleshoot Data Flow performance is to isolate transformations and sources by themselves.

While you are developing your package, you can identify bottlenecks within a specific Data Flow by making a copy of your Data Flow and begin decomposing it by replacing components with placeholder transformations. In other words, take a copy of your Data Flow and run it without any changes. This will give you a baseline of the execution time of the package. Next, remove all the Destination adapters, and replace them with the Multicast Transformation (the Multicast is a great placeholder transformation because it can act as a destination without any outputs and has no overhead).

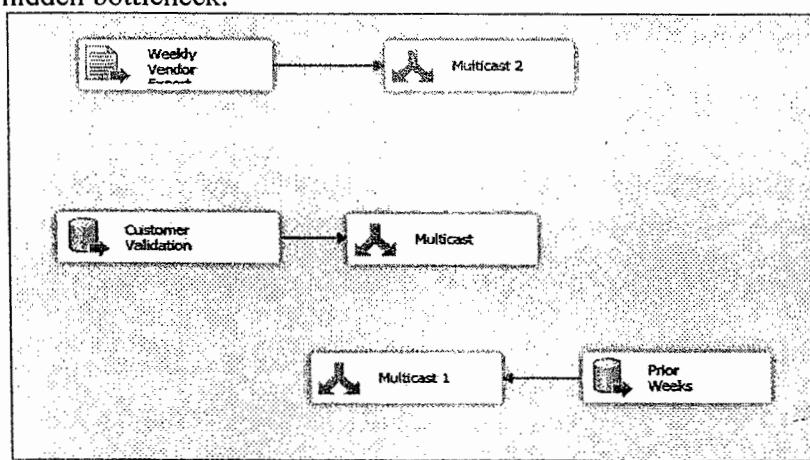
Figure 15-21 represents a modified package in which the Destination adapters have been replaced with Multicast Transformations.

If your package without the destinations still runs the same, then your performance bottleneck is a source or one of the

transformations. The next most common issue is the source, so this time, delete all your transformations and replace them

with Multicast Transformations. Run your package. If the execution time is just as slow as the first run, then you can be sure that the performance issue is one or more of the sources. If the performance is a lot faster, then you have a performance issue with one of the transformations.

Even the Active Time report shown in Figure 15-17 can give deceptive results if the source is the bottleneck. The source and all transformations in the first execution tree will show high active time. In fact, all your Data Flow components may have a high active time if you don't have any blocking transformations. Therefore, checking the sources will assist in identifying whether the Source adapter is the hidden bottleneck.



In fact, the Multicast approach can be applied repeatedly until you figure out where the issue lies. In other words, go back to your original copy of the Data Flow and replace transformations until you have identified the transformation that causes the biggest slowdown. It may be the case that you have more than one transformation or adapter that is the culprit, but with this approach you will know where to focus your redesign or reconfiguration efforts.

## PIPELINE PERFORMANCE MONITORING

Another tool available to SSIS is the Windows operating system tool called *Performance Monitor* (*PerfMon* for short), which is available to local administrators in the machine's Administrative Tools. When SSIS is installed on a machine, a set of counters is added that enables tracking of the Data Flow's performance.

The following counters are available in the SQLServer:SSIS Pipeline object within PerfMon.

Descriptions of these counters are provided next:

*BLOB bytes read*

*BLOB bytes written*

*BLOB files in use*

*Buffer memory*

*Buffers in use*

*Buffers spooled*

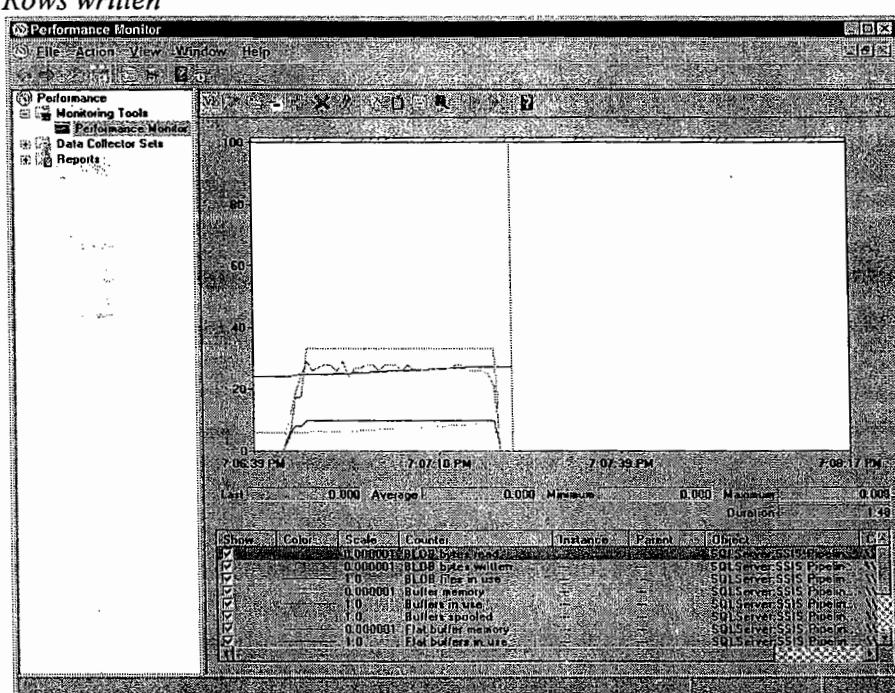
*Flat buffer memory*

*Flat buffers in use*

*Private buffers in use*

*Rows read*

*Rows written*



House

## SSIS INTERVIEW QUESTIONS AND ANSWERS

**Note: If you need assistance on these questions talk to Vinay sir.**

### Questionnaire 1

1. Difference between synchronous and asynchronous data flow transformations etc.
2. Understanding of when it's best to use script vs. the built in components.
3. Package configurations and how they work?
4. An idea of the classes and methods used in SSIS.
5. What's the difference between Control Flow and Data Flow?

6. What is the Multicast Shape used for?
7. What shape would you use to concatenate two input fields into a single output

### **Questionnaire 2**

1. What is for-loop container? Give an example of where it can be used.
2. What is foreach-loop container? Give an example of where it can be used.
3. What is sequence container? Give an example of where it can be used.
4. What is the difference between Analysis Services processing task & Analysis services execute DDL task?
5. What is the difference between for-loop container & foreach-loop container?
6. What are the different parameters or configurations that "send mail task" requires?
7. Mention few mapping operations that the Character Map transformation supports.
8. Explain the functionality of: Import Column Transformation and Export Column Transformation
9. Explain the functionality of: Percentage Sampling transformation
10. Explain the functionality of: SCD transformation
11. Explain the functionality of: Union All transformation
12. What does "Lookup" transformation used for?
13. What are checkpoints? For which objects we define checkpoint? How to configure checkpoint for a package?
14. What is the use of "package configurations" available in SSIS?
15. What are the different ways in which configuration details can be stored?
16. How to deploy a package from development server to production server?
17. How to create Integration Services Package Deployment Utility?
18. How to deploy packages to file system?
19. How to deploy packages to SQL server? Where in database packages will be stored?
20. How to set security for a package? Explain the same as per different deployment options.
21. Explain the architecture of SSIS
22. Explain the how SSIS engine workflow

Q1.SIS architecture consists of four key parts:

a) **Integration Services service:** monitors running Integration Services packages and manages the storage of packages.

b) **Integration Services object model:** includes managed API for accessing Integration Services tools, command-line utilities, and custom applications.

c) **Integration Services runtime and run-time executables:** it saves the layout of packages, runs packages, and provides support for logging, breakpoints, configuration, connections, and transactions. The Integration Services run-time executables are the package, containers, tasks, and event handlers that Integration Services includes, and custom tasks.

d) **Data flow engine:** provides the in-memory buffers that move data from source to destination.

**Q2 How would you do Logging in SSIS?**

Logging Configuration provides an inbuilt feature which can log the detail of various events like onError, onWarningetc to the various options say a flat file, SqlServer table, XML or SQL Profiler.

**Q3 How would you do Error Handling?**

A SSIS package could mainly have two types of errors

- Procedure Error: Can be handled in Control flow through the precedence control and redirecting the execution flow.
- Data Error: is handled in DATA FLOW TASK by redirecting the data flow using Error Output of a component.

**Q4 How to pass property value at Run time? How do you implement Package Configuration?**

A property value like connection string for a Connection Manager can be passed to the pkg using package configurations. Package Configuration provides different options like XML File, Environment Variables, SQL Server Table, Registry Value or Parent package variable.

**Q5 how would you deploy a SSIS Package on production?**

- Create deployment utility by setting its property as true.
- It will be created in the bin folder of the solution as soon as package is build.
- Copy all the files in the utility and use manifest file to deploy it on the Prod.

**Q6. Difference between DTS and SSIS?**

Everything except both are product of Microsoft :-).

**Q7 What are new features in SSIS 2008?**

explained in other post

<http://sqlserversolutions.blogspot.com/2009/01/new-improvementfeatures-in-ssis-2008.html>

**Q8 How would you pass a variable value to Child Package?**

too big to fit here so had a write other post

<http://sqlserversolutions.blogspot.com/2009/02/passing-variable-to-child-package-from.html>

**Q9 What is Execution Tree?**

Execution trees demonstrate how package uses buffers and threads. At run time, the data flow engine breaks down Data Flow task operations into execution trees. These execution trees specify how buffers and threads are allocated in the package. Each tree creates a new buffer and may execute on a different thread. When a new buffer is created such as when a partially blocking or blocking transformation is added to the pipeline, additional memory is required to handle the data transformation and each new tree may also give you an additional worker thread.

**Q10 What are the points to keep in mind for performance improvement of the package?**

<http://technet.microsoft.com/en-us/library/cc966529.aspx>

**Q11 You may get a question stating a scenario and then asking you how would you create a package for that e.g. How would you configure a data flow task so that it can transfer data to different table based on the city name in a source table column?****Q13 Difference between Union all and Merge Join?**

a) Merge transformation can accept only two inputs whereas Union all can take more than two inputs

b) Data has to be sorted before Merge Transformation whereas Union all doesn't have any condition like that.

**Q14 May get question regarding what X transformation do? Lookup, fuzzy lookup, fuzzy grouping transformation are my favorites. For you.****Q15 How would you restart package from previous failure point?****What are Checkpoints and how can we implement in SSIS?**

When a package is configured to use checkpoints, information about package execution is written to a checkpoint file. When the failed package is rerun, the checkpoint file is used to restart the package from the point of failure. If the package runs successfully, the checkpoint file is deleted, and then re-created the next time that the package is run.

**Q16 Where are SSIS package stored in the SQL Server?**

MSDB.sysdtspackages90 stores the actual content and sysdtscategories, sysdtslog90, sysdtspackagefolders90, sysdtspackagelog, sysdtssteplog, and sysdtstasklog do the supporting roles.

**Q17 How would you schedule a SSIS packages?**

Using SQL Server Agent. Read about Scheduling a job on Sql server Agent

**Q18 Difference between asynchronous and synchronous transformations?**

Asynchronous transformation have different Input and Output buffers and it is up to the component designer in an Async component to provide a column structure to the output buffer and hook up the data from the input.

**Q19 How to achieve parallelism in SSIS?**

Parallelism is achieved using MaxConcurrentExecutable property of the package. Its default is -1 and is calculated as number of processors + 2.

# **REPORTING SERVICES**

# **Vinay Tech House**

## ADD / ENHANCED RS FUNCTIONALITY IN 2012

As you work through the book, we will point out the enhancements released with SSRS 2008 R2 and SSRS 2012, remaining get it from Vinay Sir.

### NEW RENDERERS

#### **Excel 2010 Renderer**

By default, the Excel rendering option now produces an XLSX file in Open Office XML format, which you can open in either Excel 2007 or Excel 2010 if you have the client installed on your computer. The benefit of the new file type is the higher number of maximum rows and columns per worksheet that the later versions of Excel support—1,048,576 rows and 16,384 columns. You can also export reports with a wider range of colors as well, because the XLSX format supports 16 million colors in the 24-bit color spectrum. Last, the new renderer uses compression to produce a smaller file size for the exported report.

#### **Word 2010 Renderer**

Although the ability to render a report as a DOCX file in Open Office XML format does not offer as many benefits as the new Excel renderer, the new Word render does use compression to generate a smaller file than the Word 2003 renderer. You can also create reports that use new features in Word 2007 or Word 2010.

## SHAREPOINT SHARED SERVICE ARCHITECTURE

#### **Shared Service Architecture Benefits**

With Reporting Services available as a shared service application, you can experience the following new benefits:

- Scale Reporting Services across web applications and across your SharePoint Server 2010 farms with fewer resources than possible in previous versions.
- Use claims-based authentication to control access to Reporting Services reports.
- Rely on SharePoint backup and recovery processes for Reporting Services content.

TABLE 10-1 SharePoint Edition Feature Support

Reporting Services Feature	SharePoint Foundation 2010	SharePoint Server 2010 Standard Edition	SharePoint Server 2010 Enterprise Edition
General report viewing and subscriptions	✓	✓	✓
Data Alerts			✓
Power View			✓

### \*\* POWER VIEW \*\*

Power View is the latest self-service feature available in Reporting Services. It is a browser-based Silverlight application that requires Reporting Services to run in SharePoint integrated mode using SharePoint Server 2010 Enterprise Edition. It also requires a specific type of data source—either a tabular model that you deploy to an Analysis Services server or a PowerPivot workbook that you deploy to a SharePoint document library.

**Business Intelligence Semantic Model (BISM) connection file** Create a BISM file to Connect to either a PowerPivot workbook or to an Analysis Services tabular model. You can use this file as a data source for both Power View reports and Excel workbooks

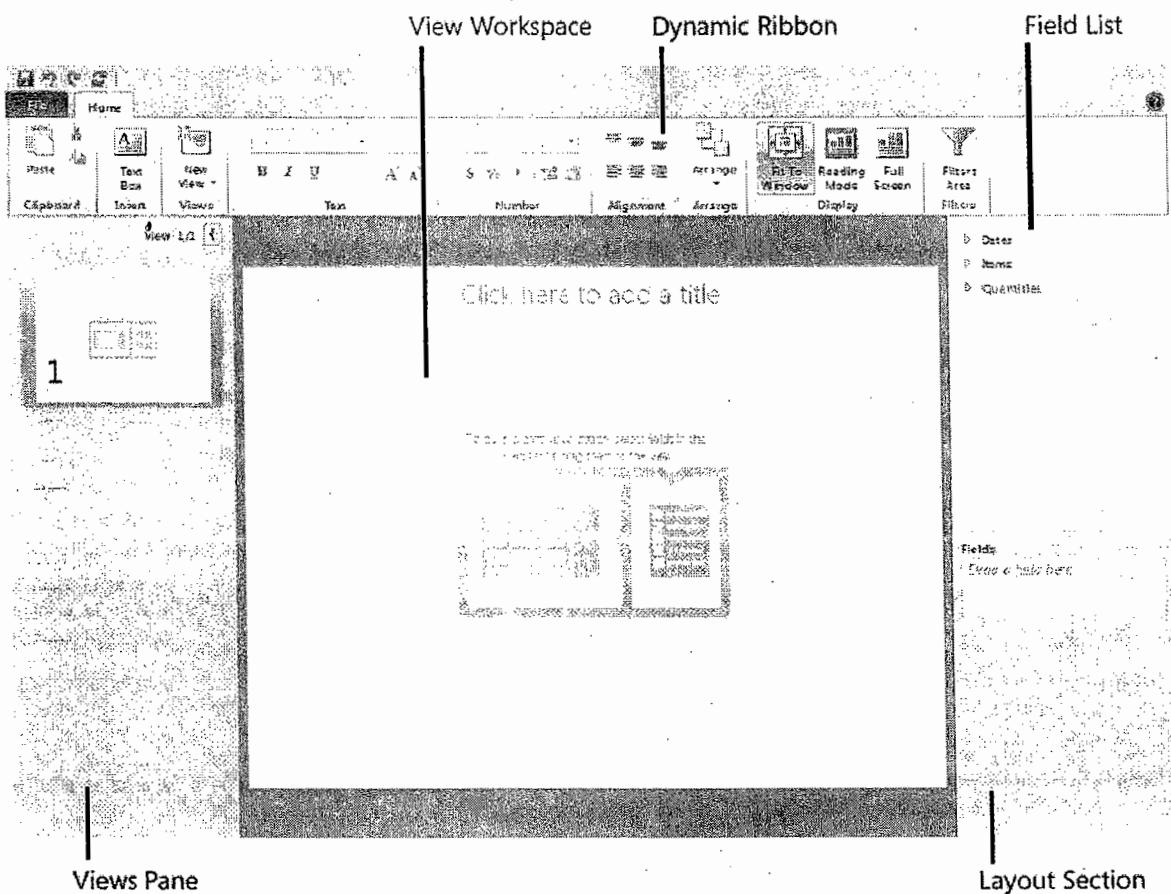


FIGURE 10-2 Power View design environment

#### a) Data Visualization

A table is only one way to explore data in Power View. After you add several fields to a table, you can then convert it to a matrix, a chart, or a card. If you convert it to a scatter chart, you can add a play axis to visualize changes in the data over multiple time periods. You can also create multiples of the same chart to break down its data by different categorizations.

### Cards and Tiles

Another type of visualization you can use is *cards*, which is a scrollable list of grouped fields arranged in a card format, as shown in Figure 10-7. Notice that the default label and default image fields are more prominent than the other fields. The size of the card changes dynamically as you add or remove fields until you resize using the handles, after which the size remains fixed. You can double-click the sizing handle on the border of the card container to revert to auto-sizing.

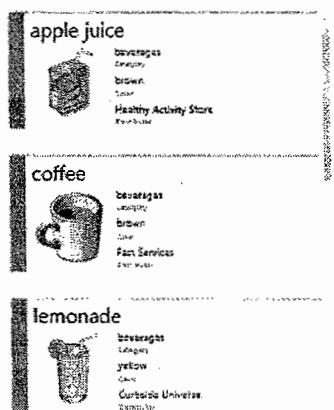


FIGURE 10-7 Card visualization

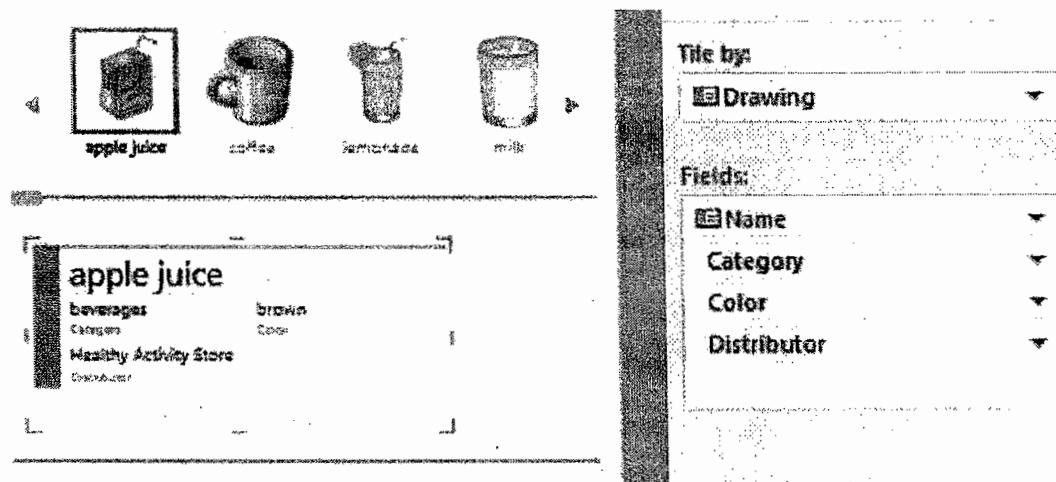


FIGURE 10-8 Tiles

### b) Play Axis

When your table or chart contains two measures, you can convert it to a scatter chart to show one measure on the horizontal axis and the second measure on the vertical axis. Another option is to use a third measure to represent size in a bubble chart that you define in the layout section, as shown in Figure 10-9. With either chart type, you can also add a field to the Color section for grouping purposes. Yet one more option is to add a field from a date table to the Play Axis area of the layout section.

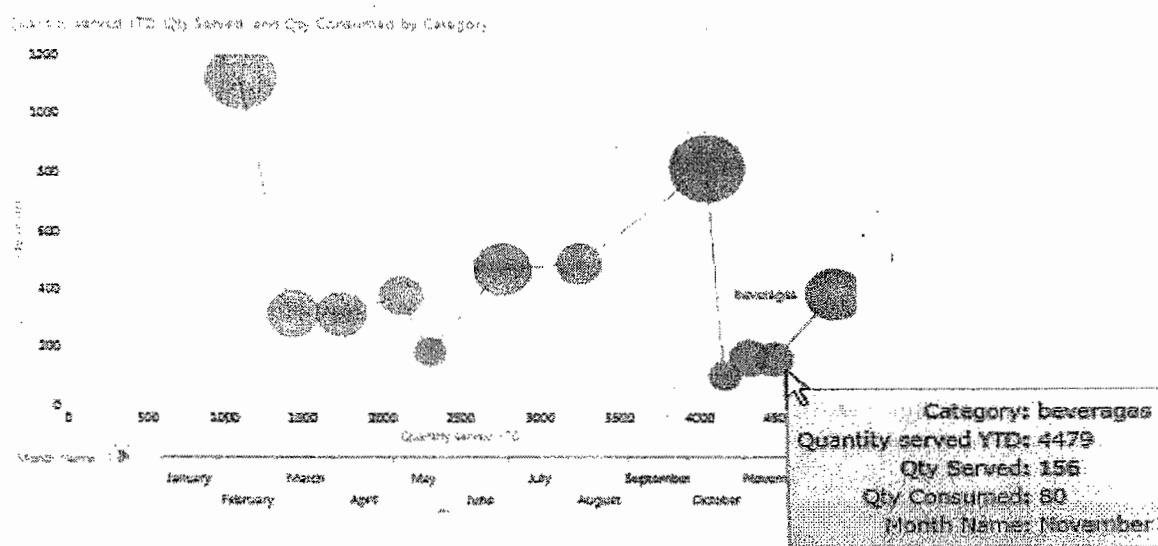


FIGURE 10-10 Bubble chart with a play axis

### c) Data Alerts

Rather than create a subscription to email a report on a periodic basis, regardless of the data values that it contains, you can create a data alert to email a notification only when specific conditions in the data are true at a scheduled time. This new self-service feature is available only with Reporting Services running in SharePoint integrated mode and works as soon as you have provisioned subscriptions and alerts. However, it works only with reports you create by using Report Designer or Report Builder. You cannot create data alerts for PowerView reports.

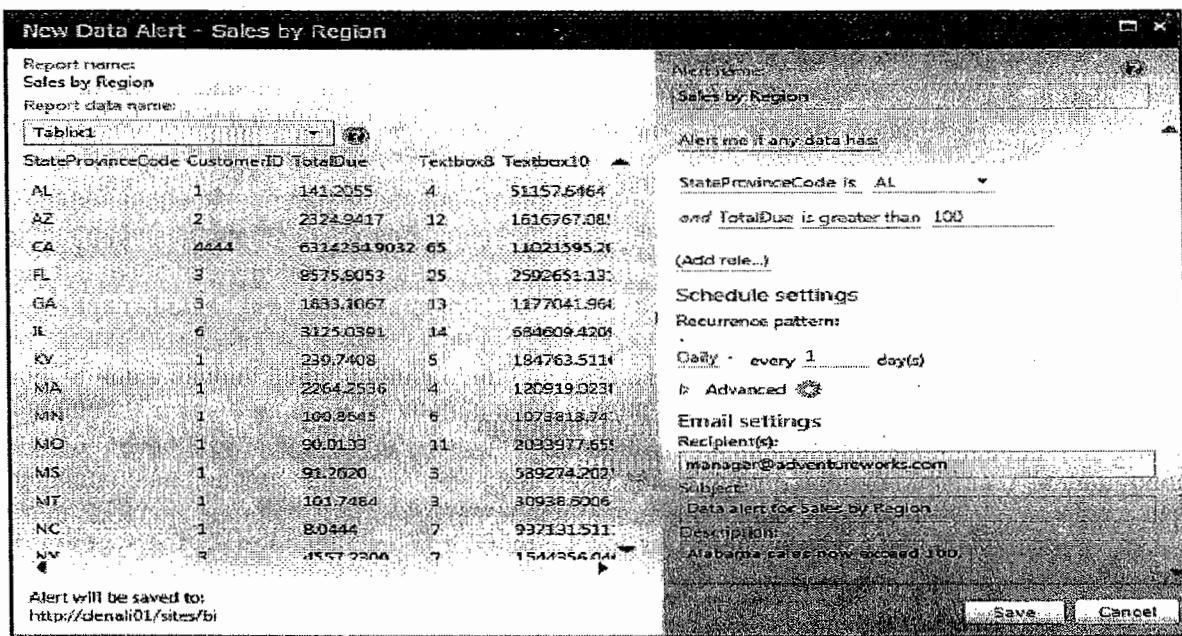
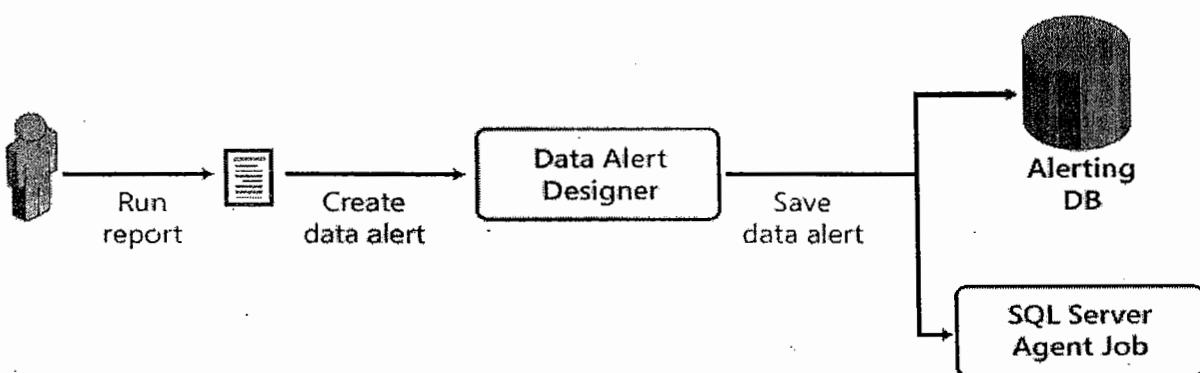


FIGURE 10-20 Data Alert Designer

**FIGURE 10-21** Data alert creation process

### Schedule settings

Recurrence pattern:

Daily - every 1 day(s)

Advanced

Start alert on:

Stop alert on:

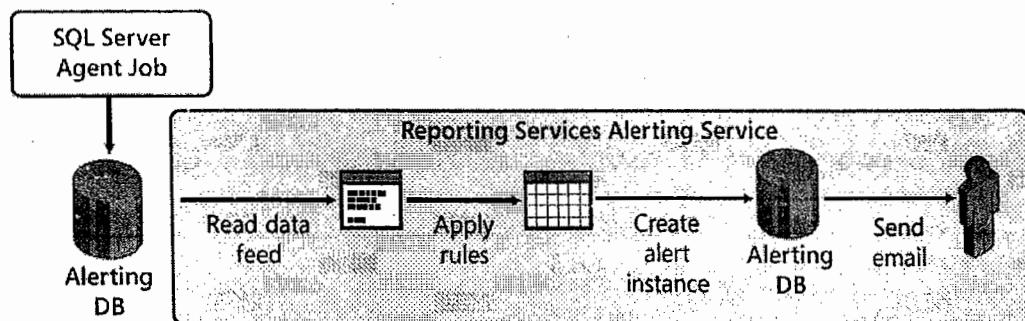
Send message only if alert results change

**FIGURE 10-22** Data alert advanced schedule settings

OUSE

### Alerting Service

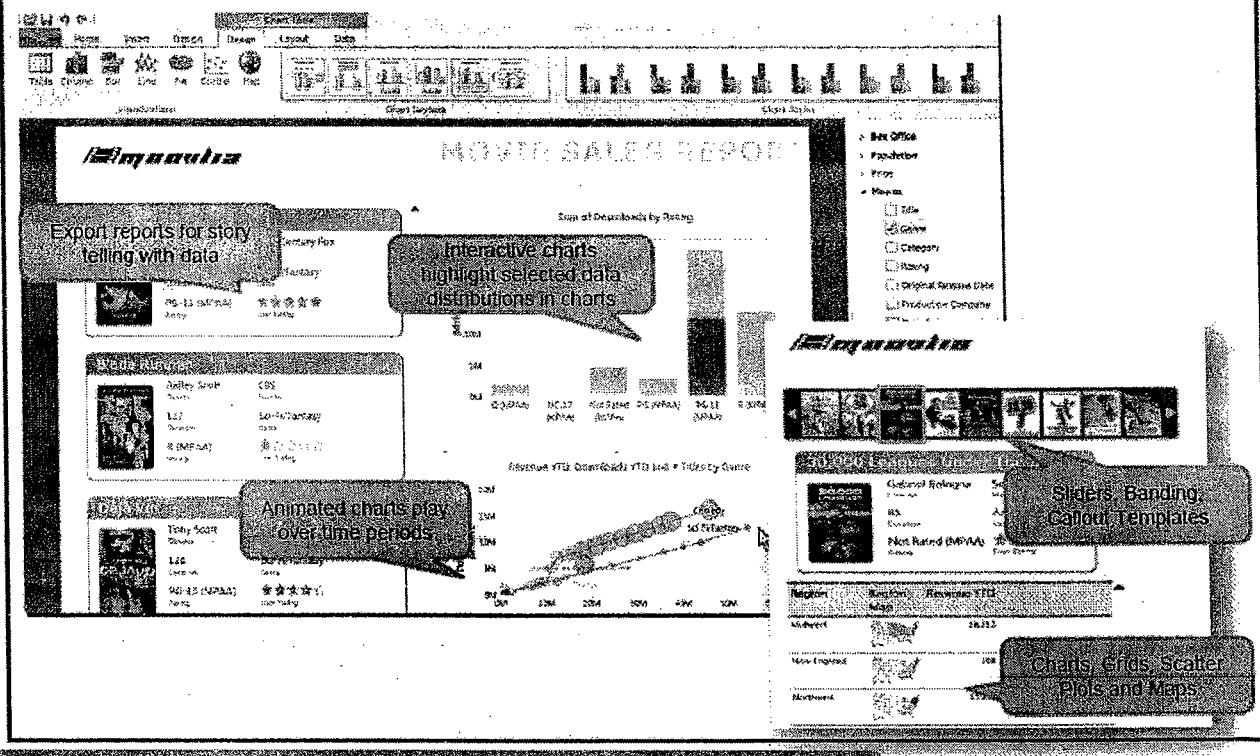
The Reporting Services alerting service manages the process of refreshing the data feed and applying the rules in the data alert definition, as shown in Figure 10-23. Regardless of the results, the alerting service adds an alerting instance to the alerting database to record the outcome of the evaluation. If any rows in the data feed satisfy the conditions of the rules during processing, the alerting services generate an email containing the alert results.

**FIGURE 10-23** Alerting service processing of data alerts

## New Project Crescent

- Simple to use interactive, visual data exploration
  - Ad-hoc reporting off PowerPivot and BISM models in SharePoint
  - Analysis Services is a high priority on the roadmap!
- Highly visual design experience
  - Allows users to visualize and interact with data directly in a web browser
  - Quick and simple drag-drop highly interactive reporting
  - Object templates for easy report design
  - Contextual selections on reports without any programming
  - Slicers, Basic and Advanced shown or hidden filtering
  - Variety of visualization types, vertical and horizontal multiples
- Presentation-ready at all times
  - Designed for story telling with data
  - “Play” reports over time periods to view data changes over time

## New Project Crescent



## \*\* RS FEATURES DEPRECATED IN SQL SERVER 2012

*functionality reduce*

### 1. HTML Rendering Extension Device Information Settings

The following device information settings for the HTML rendering extension are deprecated.

- ActionScript
- ActiveXControls
- GetImage
- OnlyVisibleStyles
- ReplacementRoot
- ResourceStreamRoot
- StreamRoot
- UsePx
- Zoom

### 2. Microsoft Word and Microsoft Excel 1997-2003 Rendering

The Reporting Services BIFF8 rendering extensions Reporting Services reports to the Microsoft Word and Microsoft Excel 1997-2003 binary interchange file format. SQL Server 2012 includes extensions that render in the Microsoft Office 2007-2010 Open XML format.

#### **Report Definition Language (RDL) 2005 and Earlier**

**Vinay Tech House**  
Report Definition Language (RDL) 2005 and earlier is deprecated.

#### **SQL Server 2005 and earlier Custom Report Items**

Custom Report Items (CRI) compiled for SSRS 2005 and earlier are deprecated.

#### **Reporting Services Snapshots 2005 and earlier**

RS snapshots rendered with SQL Server Reporting Services 2005 and earlier are deprecated.

#### **Report Models**

Semantic modeling language (SMDL) report models are deprecated. Although you can continue to use existing report models as data sources in SQL Server 2012 Reporting Services reports you should consider updating your reports to remove their dependency on report models. SQL Server 2012 Reporting Services does not include tools for creating or updating report models.

#### **Deprecated Methods in the Web Service Endpoint**

The following operations are deprecated in the ReportingService2010 Web service endpoint:

- GetProperties(String, Property[])      IsSSLRequired()

**\*\* SQL SERVER 2012 RS BEHAVIOUR CHANGES \*\*****View Items permission will not download Shared Datasets (SharePoint Mode)**

New Behavior: Users with the SharePoint permission of "View Items" can no longer download the contents of Reporting Services shared datasets. This behavior change is now consistent with the "View Items" permissions for reports, data sources, and models. Users with "View Items" permission can view and execute reports, data sources, and models but they cannot download their content. Previous Behavior: Users with the "View Items" SharePoint permission could download the contents of Reporting Services shared datasets.

**Report Server trace logs are in a new location for SharePoint mode (SharePoint Mode)**

New behavior: For a report server installed in SharePoint mode, the report server trace logs will be under %Programfiles%\Common Files\Microsoft Shared\Web Server Extensions\14\Web Services\ReportServer\LogFiles.

Previous Behavior: Report Server trace logs were found under a path similar to the following:  
%Programfilesdir%\Microsoft SQL Server\<RS\_instance>\Reporting Services\LogFiles

**GetServerConfigInfo SOAP API is no longer supported (SharePoint Mode)**

New behavior: Use PowerShell cmdlet Get-SPRSServiceApplicationServers

Previous Behavior: Customers could develop SOAP client code to communicate directly with the Reporting Services end point, and call GetReportServerConfigInfo().

**Report Server Configuration and Management Tools*****Configuration Manager is not used for SharePoint Mode***

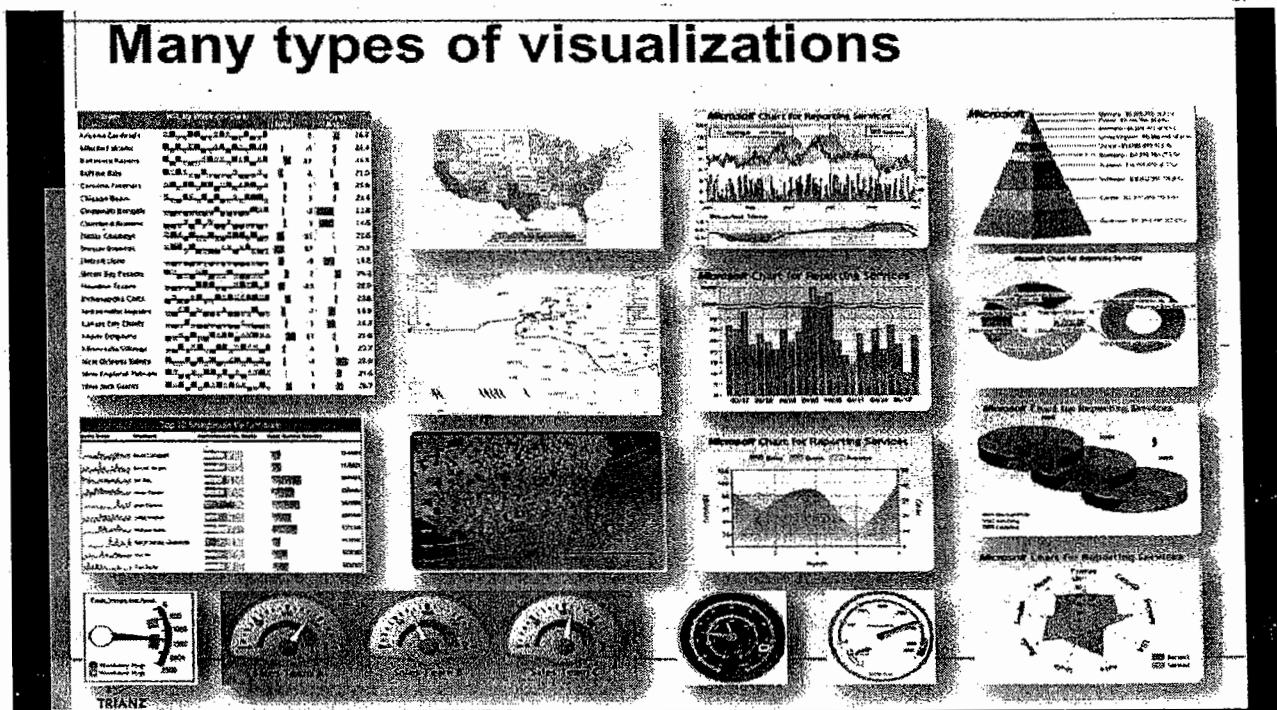
New behavior: The Reporting Services Configuration Manager no longer supports SharePoint Mode report servers. Configuration of Reporting Services SharePoint mode can now be completed by using SharePoint Central administration and therefore Reporting Services Configuration Manager no longer supports SharePoint mode. Configuration Manager is now only used for Native mode report servers.

**You cannot change the server from one mode to another:**

New behavior: You cannot change server modes. If you install a report server as Native mode you cannot change or re-configure it to be SharePoint mode. If you install in SharePoint mode, you can change the report server to native mode.

Previous Behavior: Customer installs a Reporting Services report server in SharePoint mode. If the customer wants to switch the report server to Native mode, they could open the Reporting

Services configuration manager to switch to Native mode by either creating a new or connecting to an existing Native mode database. The customer could also use Reporting Services Configuration Manager to switch from SharePoint mode to Native mode.



## Vinay Tech House

Each release of Reporting Services since its introduction in Microsoft SQL Server 2000 has expanded its feature base to improve your options for sharing reports, visualizing data, and empowering users with self-service options. SQL Server 2012 Reporting Services is no exception, although almost all the improvements affect only SharePoint integrated mode. The exception is the two new renderers available in both native mode and SharePoint integrated mode.

Reporting Services in SharePoint integrated mode has a completely new architecture, which you now configure as a SharePoint shared service application. For expanded data visualization and self-service capabilities in SharePoint integrated mode, you can use the new ad reporting tool, Power View. Another self-service feature available only in SharePoint integrated mode is data alerts, which allows you to receive an email when report data meets conditions you specify.

### SSRS INSTALLATION MODES AND OPTIONS

Reporting Services Native Mode

Reporting Services Sharepoint Integrated mode

Note: Mode indicates the databases and configurations in the created area.

## SSRS OPPORTUNITIES

- a) SSRS developers.
- b) MSBI developer with 6 months of experience.
- c) SQL server developer /DBA/.net developer + SSRS.

## SSRS DEVELOPER RESPONSIBILITIES

1. Read the content on first page.
2. Creating /implementing the below points on the report
  - a) applying parameters
  - b) creating actions
  - c) creating snapshots, caches etc.
  - d) creating calculations
  - e) Applying formations and using various report items like regions
  - f) Applying methods for fine tuning the reports etc..

## SSRS DBA RESPONSIBILITIES

1. Configuration and set up maintenance.
2. Implementing role based security.
3. Working on system bottlenecks etc..

## SSRS MAINTENANCE / SUPPORT RESPONSIBILITIES

1. Creating modifying subscribing (or) jobs
2. working on failed jobs
3. Advancing the jobs etc.

## NEED OF REPORTS

The data whatever loaded through ETL (Such as a SSIS), the data whatever analyzed (SSAS) to present understandable format to the end user and analysts we need a report.

For doing this we use various reporting tools such as SSRS, COGNOS, BO, MicroStrategy, OBIEE, Hyperion etc...

Reports are used generally two purposes

- a) Internal b) External
  - a) **Internal:** - Company internal operations such as pay slips, Experience Letter, etc.,
  - b) **External:** - For submitting to a report to third party authorities like income tax, STPI (software technology part income).

There are three ways to create the reports

- a) Standard Reporting
- b) Adhoc Reporting
- c) Embedded Reporting

i. **Standard Reports:** - From central repository the reports are generated, users can take

whatever reports they need.

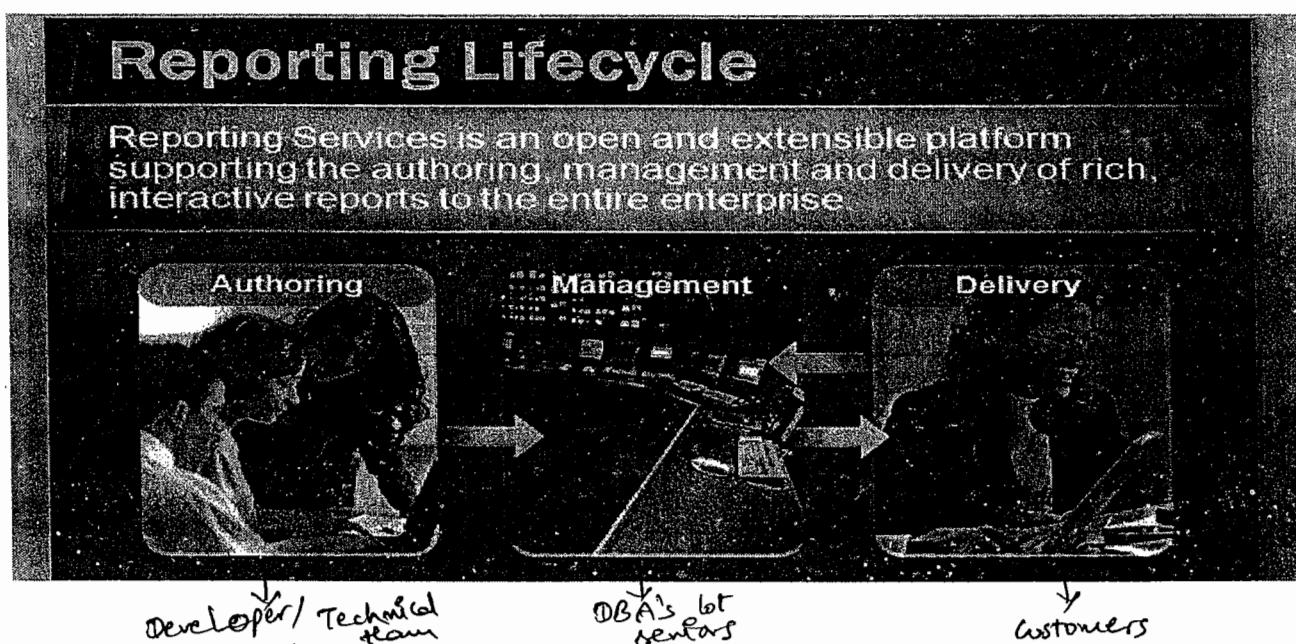
ii. **Adhoc Reports:** -

Depending on the situation, if you create a report then it's called adhoc report.

iii. **Embedded Reports:** -

Embedding the report to any third party application such as (.Net, Java, etc.,).

## REPORTING SERVICES LIFE CYCLE



## Full Reporting Life Cycle Support

- **Authoring:** Report developers can create reports to be published to the Report Server using Microsoft or third-party design tools that use Report Definition Language (RDL), an XML-based industry standard used to define reports.
- **Management:** Report definitions, folders, and resources are published and managed as a Web service. Managed reports can be executed either on demand or on a specified schedule, and are cached for consistency and performance.
- **Delivery:** Supports both on-demand (pull) and event-based (push) delivery of reports. Users can view reports in a Web-based format or in e-mail.

### \*\* SSRS ARCHITECTURE AND DIFFERENT DIAGRAMS

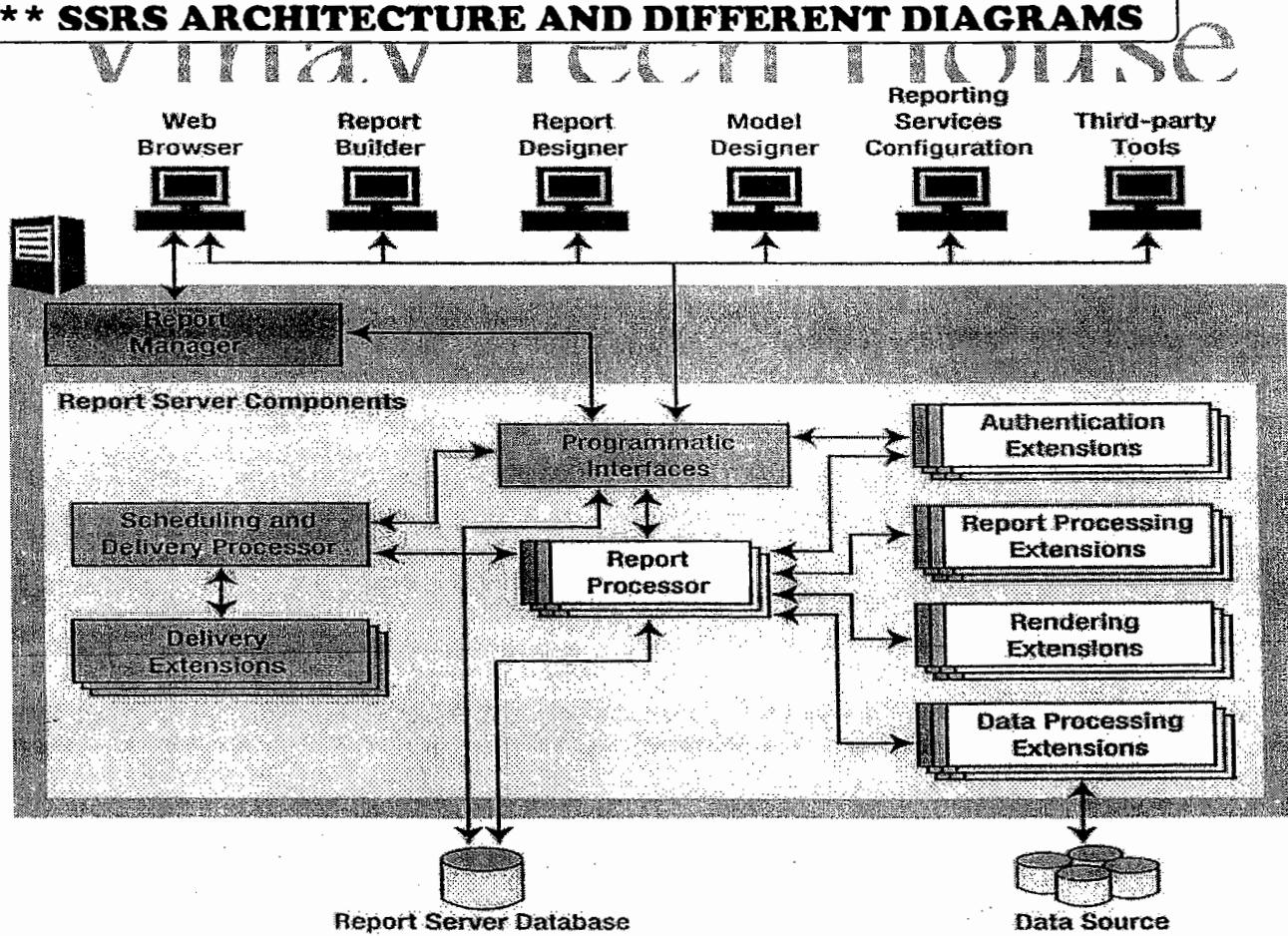


Figure 1-1 shows the conceptual breakdown of the three component pieces.

The data source and the SSRS databases, ReportServer and ReportServerTempDB, are separate entities.

## DATA SOURCE

The **data source** is the origin of the data that will populate the reports, while the report server databases store Metadata and execution information about the reports. The data source can be any supported data provider, such as SQL Server, Oracle, Lightweight Directory Access Protocol (LDAP), Microsoft SharePoint List, SQL Azure and Analysis Services.

## SSRS Databases

SSRS is added as an option during the SQL Server installation process. The SSRS native installation creates two databases that are used to store report metadata and manage performance:

### ReportServer:

This is the primary database that stores all the information about reports that was originally provided from the RDL files used to create and publish the reports to the ReportServer database. In addition to report properties (such as data sources and report parameters), ReportServer also stores folder hierarchy and report execution log information.

### ReportServerTempDB:

This database houses cached copies of reports that you can use to increase performance for many simultaneous users. By caching reports using a nonvolatile storage mechanism, you make sure they remain available to users even if the report server is restarted.

**Note:** An additional database might be added after the initial installation of SSRS: the **RSExecutionLog** database. This stores more detailed information about report execution, such as the user who ran the report, the time of execution, and performance statistics.

**Note :** When configuring Reporting Services to run in SharePoint integrated mode for 2012, an extra database is installed for **Alerting**. Note also that the default database names are slightly different as they will have a unique identifier appended to ReportingService\_ assigned when creating the instance of Reporting Services on the SharePoint site. For example, in SharePoint integrated mode, **ReportServerTempDB becomes something like ReportingService\_14214aae2b5d4f0d888289011932bmcdTempDB.**

The SSRS Report Server The SSRS report server plays the most important role in the SSRS model. Working in the middle, it's responsible for every client request to render a report or to

perform a management request, such as creating a subscription. You can break down the report server into several subcomponents by function:

- Programming interface
- Authentication Layer (new in SSRS 2008)
- Report processing
- Data processing
- Report rendering
- Report scheduling and delivery

## SSRS Web Service Interface

The programming interface, exposed as .NET Web service application programming interfaces (APIs) and uniform resource locator (URL) access methods, handles all incoming report and management requests from clients. Depending on the type of request, the programming interface either processes it directly by accessing the ReportServer database or passes it off to another component for further processing. If the request is for an on-demand report or a snapshot, the Web service passes it to the Report Processor before delivering the completed request to the client or storing it in the ReportServer database.

## Authentication Layer

SSRS 2005 relied heavily on the authentication methodology of Internet Information Services (IIS), since SSRS and IIS were interdependent. With the exception of SSRS in SharePoint integrated mode, no SSRS versions since 2008 are tied to IIS. SSRS now uses Http.sys directly, as well as SQL Server's native network components, so SSRS's architecture has been redesigned to include its own authentication layer,

## The Report Processor

The Report Processor component is responsible for all report requests. Like the programming interface, it communicates directly with the ReportServer database to receive the report definition information that it then uses to combine with the data returned from the data source, which is accessed via one of the data processing extensions.

## Data Processing

Reporting Services supports twelve data processing extensions to connect to data sources.

These are:

- SQL Server
- Oracle
- OLE DB
- OLEDB-MD
- ODBC
- XML

- SAP BI NetWeaver
- Hyperion Essbase
- Teradata
- Microsoft SQL Azure (SQL in the Cloud)
- Microsoft SQL Server Parallel Data Warehouse
- Microsoft SharePoint List

When the data processing component receives the request from the Report Processor, it initiates a connection to the data source and passes it the source query. Data is returned and sent back to the Report Processor, which then combines the elements of the report with the data returned from the Data Processor extension.

## Report Rendering

The combined report and data is handed off to the rendering extension component to be stored in an intermediate format called Report Page Layout (RPL). The RPL is then turned into one of several supported or custom formats, based on the rendering type specified by the client **HTML**: Default rendering format, supporting HTML versions 4.0 and 3.2.

**Portable Document Format (PDF):** Format used to produce print-ready reports using Adobe Acrobat Reader. SSRS doesn't require that you have an Adobe license to render in PDF, which is a great benefit to customers. All you need is a PDF reader.

**Excel 2002 and 2003:** Service Pack 1 of SSRS supports Excel 97 and later. As discussed previously, SQL Server 2012 supports exporting in 2007-2010 (.xlsx) compressed format to allow more rows and columns.

**XML:** Other applications or services can use reports that are exported to XML.

• **Comma-separated values (CSVs):** By rendering to a CSV file, you can further process the report by importing it into other CSV-supported applications such as Microsoft Excel.

**MIME HTML (MHTML):** You can use this format, also known as a *Web archive*, to deliver reports directly in e-mail or to deliver them for storage, because the report contents, including images, are embedded within a single file.

**Tagged Image File Format (TIFF):** Rendering image files using TIFF guarantees a standard view of the report, as it's processed the same way for all users, whatever their browser settings or versions.

**Microsoft Word:** Standard Microsoft Word document export is now included in SSRS 2008. Both 97-2003 (.doc) and 2007-2010 (.docx) compressed formats are available in SSRS 2012.

**ATOM:** This format can be consumed by ATOM-compliant client applications such as PowerPivot and SharePoint.

**NULL:** The NULL rendering extension isn't actually a format like the others, but can be used to cache results of reports. The next time the report is requested; it is pulled from cache and rendered significantly quicker. This is especially useful if you have a larger report that takes an abnormally long time to render. You will see this extension as a delivery format when creating subscriptions.

## Scheduling and Delivery

If the request from the client requires a schedule or delivery extension, such as a snapshot or

subscription, the programming interface calls the Scheduling and Delivery Processor to handle the request. You can generate and deliver report snapshots, based on a user-defined or shared schedule, to one of two supported delivery extensions: an e-mail or a file share. Note that SSRS uses the SQL Server Agent to create the scheduled job. If the SQL Server Agent isn't running, the job won't execute.

### **Reporting Services Configuration Manager:**

SSRS for SQL Server 2008 included an enhanced Reporting Services Configuration Manager designed specifically to change many of these properties in a graphical environment, including setting up the SSRS environment for offline or disconnected reporting.

When thinking of a Web-based application, the natural inclination is to think *Web browser*. Even though other front-end tools, such as SSMS and Visual Studio, connect to the report server, a Web browser plays an important role in providing the graphical interface for users, who can use Report Manager to view or print reports or remotely manage the report server for their workgroups or departments.

## **\*\* INSTALLING AND CONFIGURING \*\***

You can install Reporting Services—like Analysis Services, Integration Services, and Notification Services—as part of the main SQL Server installation. Before you install Reporting Services 2012, you must know whether you are going to run in native or SharePoint integrated mode. With earlier releases of Reporting Services, you could change mode after installation by using the **Reporting Services Configuration Manager**, but with the 2012 release, you can only **configure** Reporting Services in SharePoint integrated mode at SQL Server installation, as the application will be installed and configured as a native SharePoint service.

The components installed depend on the mode your organization chooses to implement. For instance, when installing in native mode, components include the Reporting Services Web service, ReportServer databases, and Report Manager web application. Once you have installed Reporting Services, you can disable some features if you do not need them. You might choose to do this, for example, if you wanted to use a custom application to interface with the ReportServer rather than the built in Report Manager web application. Installed client components include the administrative command-line tools mentioned previously, such as rs and rsconfig, as well as documentation, samples, and the Reporting Services Configuration Manager mentioned earlier.

If you use the default configuration settings for native mode during the installation of Reporting Services, you will use the RS Configuration Manager to set up or change various other settings related to the instance. Some tasks to perform are:

- Choosing security settings, such as whether the report server will use HTTP, HTTPS, or both.
- Configuring a Simple Mail Transfer Protocol (SMTP) mail server to handle the delivery of subscriptions.
- Backup and restore the encryption key used when you move a report server to a new machine.

• Set up Web Service and Report Manager URLs and define the ports they run on. This gives you the ability to run multiple reporting services instances on one machine. As mentioned, once you have installed SSRS, you can modify some of the configuration settings. For example, after reviewing performance data, you might decide that the report server needs to connect to an existing Web farm. You can perform this task using the **rsconfig utility or by using the graphical Reporting Services Configuration Manager**. You can reconfigure the security settings or the mail server by directly modifying the RSReportServer.config file. We'll cover using these tools, modifying the configuration file settings, and gathering performance measures. If you choose to use SharePoint integrated mode, the configuration settings are managed completely within SharePoint's Central Administration.

## Deploying SSRS Securely

SSRS is a role-based application that provides access to the objects it stores through the use of defined roles, such as content browsers who may only view reports and report data. The roles are associated with Windows-based login accounts, so SSRS relies on Windows as its primary source of authentication. It is possible to extend the security model for SSRS to support other methods of authentication, such as forms-based authentication, whereby users can log in with accounts maintained outside Windows to access the report server. Since SSRS has multiple authentication points—namely, at the report server level through http.sys and the data-access level, SQL or Windows authentication—specific security risks exist when altering the default Windows roles-based security model.

# Vinay Tech House

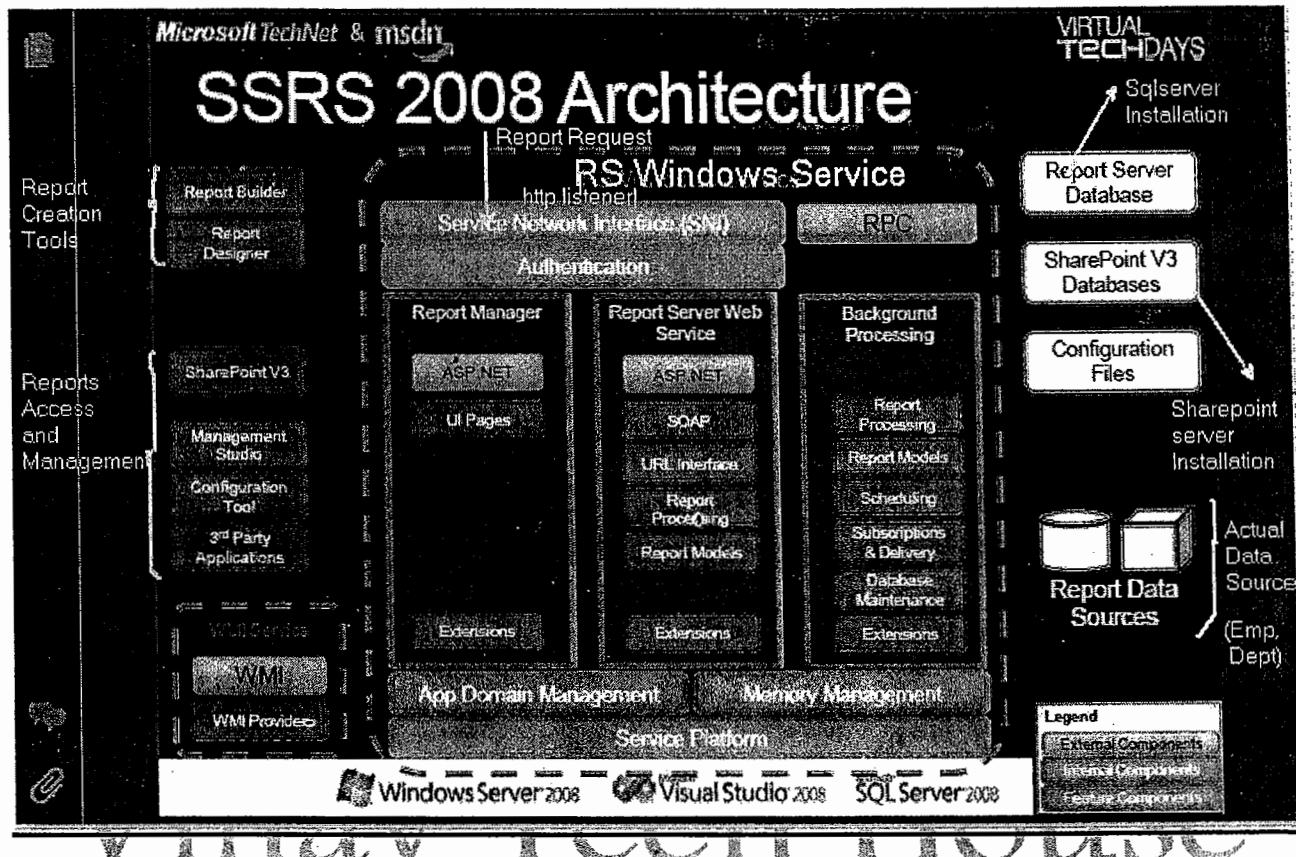
Another risk is that SSRS can support only one security extension at a time. In other words, a single SSRS report server can be extended to support a non-default authentication model or remain as a default Windows authentication, but cannot take advantage of both models simultaneously. Depending on your level of need for custom security—say, for example, you need to deploy SSRS on an Internet-facing server, or your application already supports forms authentication, and it would be too difficult to work within the constraints of Windows authentication—then you might need to consider a custom security extension. Our needs were such that we could easily incorporate SSRS into an existing Windows authentication model.

Another method of dealing with security is through a Windows or Web-based application that has its own authentication layer. Using the ReportViewer control within the Visual Studio designer for Windows and Web forms, you can use an application as a portal into the report server. As long as the application handles security, you can give the application server access to the needed objects within Reporting Services by using an Active Directory computer account like DOMAIN\ServerName\$.

### Two deployment scenarios:

- **Intranet deployment using Virtual Private Network (VPN) and firewall technologies to allow access to the SSRS report server**
- **An Internet-hosted application that uses Terminal Services to connect securely to an SSRS report server**

## **SSRS VARIOUS COMPONENTS**



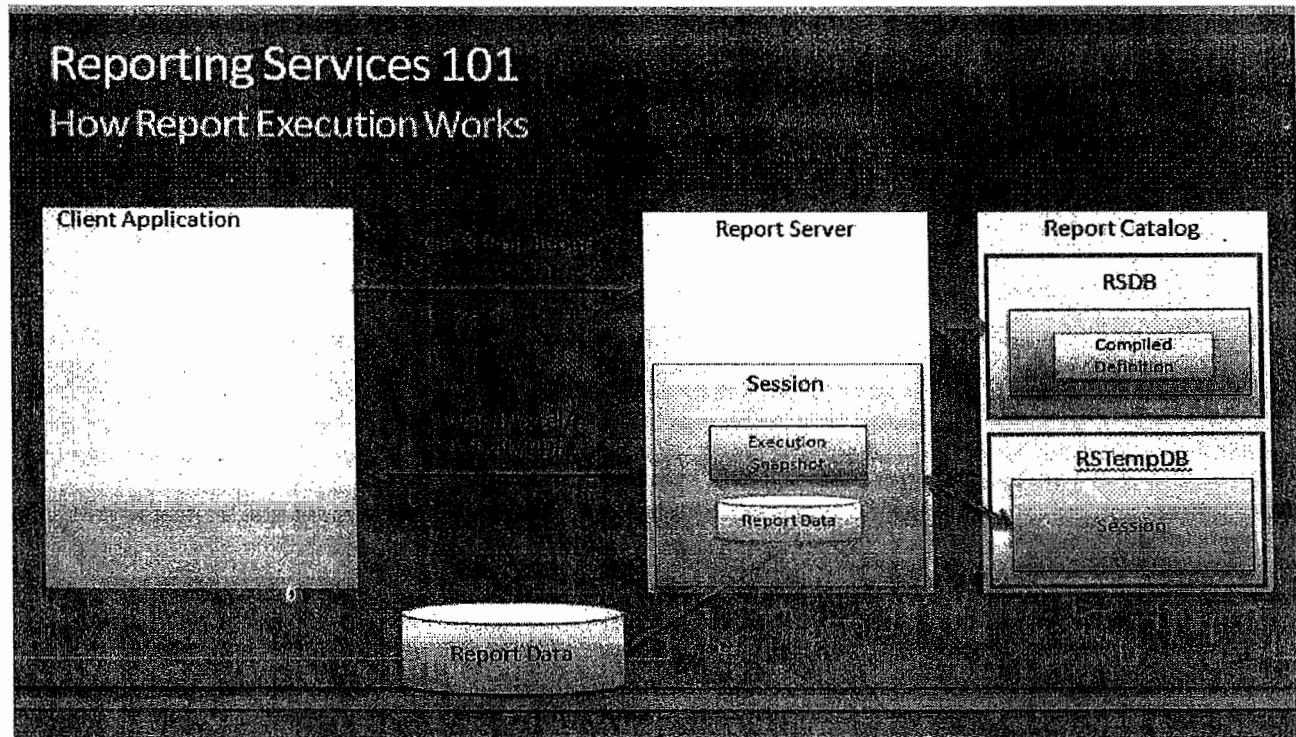
### **HTTP Listener:-**

- It maintains incoming request and directed to HTTP.SYS on a specific port on the local Computer.
- It processes requests to the authentication layer to verify the user identify.

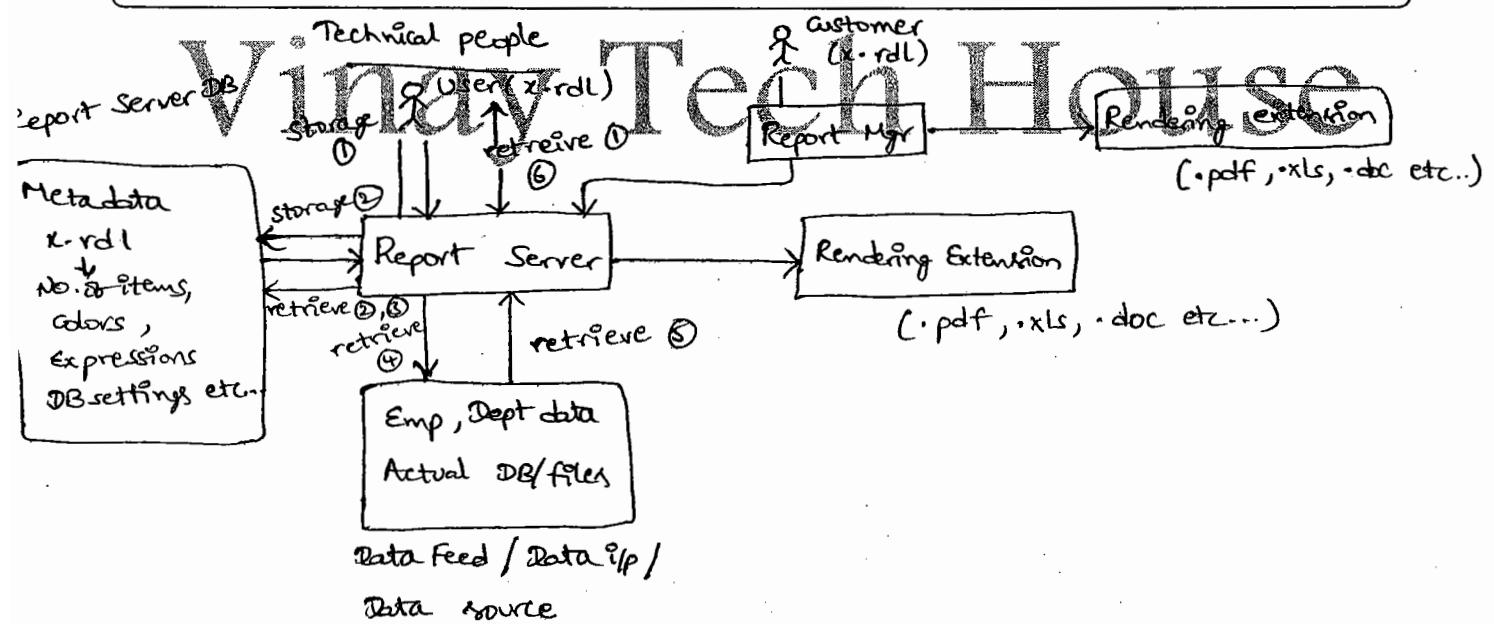
### **Authentication Layer:-**

It verifies the identity of the user or application that makes the request.

- It supports the below authentication
  - i. Windows integrated security
  - ii. NTLM Authentications
  - iii. Forms (Custom) Authentications
  - iv. Basic Authentications



### SSRS SIMPLE ARCHITECTURE (VINAY SIR APPROACH)



*Drag & Drop*

## WORKING WITH REPORTING SERVICES CONFIGURATION(DBA)

- When we install reporting services it will keeps all the received configurations to operate in configuration files.
- To monitor or manage there are 2 methods available ① opening configuration files and changing the settings ② To change some imp settings there is a flexible GUI component (Reporting services configuration mgr).

Configuration files location: Generally in the installed drive (C/D/E etc...)

program files → Microsoft Sql server → MSRS//instance name/ Reporting Service

- RS webapplication.config → Report Mgr settings available
- RS Report Designer.config → Settings related to custom data processing & render
- RS Report Server.config → Here max. possible settings related to report server, my email settings etc.. available.

# Vinay Tech House

## **Reporting Services Configuration Manager:**

Useful for two operations.

1. To monitor the component settings.
2. To manage the component settings.

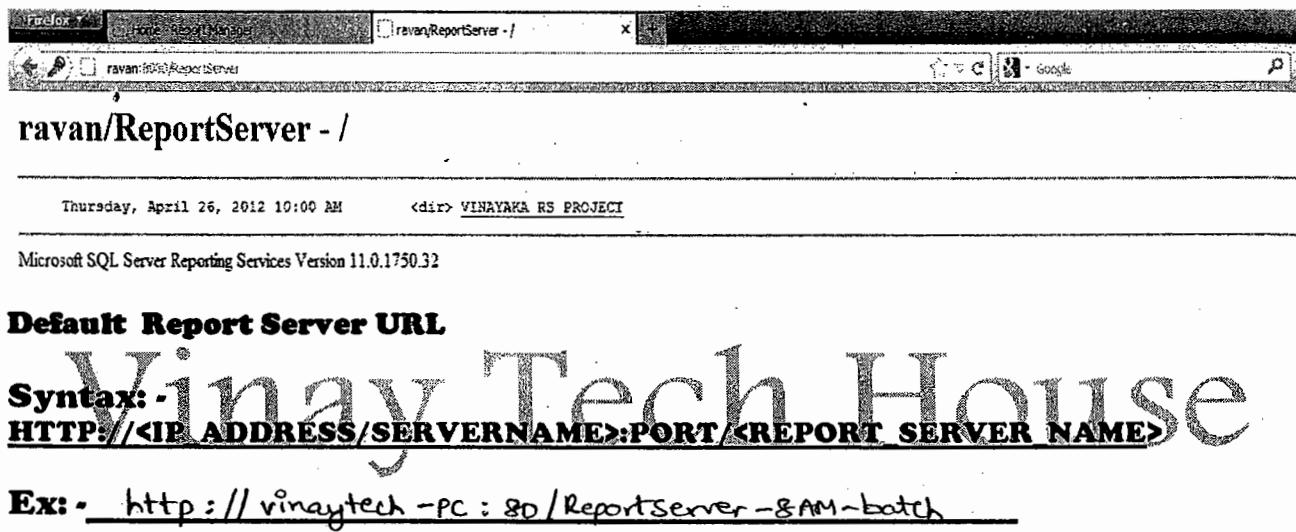
### **Navigation:**

START → Programs → SQL Server 2012 → Configuration Tools → Reporting Services Configuration manager

### **Service Account:**

Specify a built in account or windows domain user account to run the reporting server services.

**Web Server URL:** - This is report server URL



### **Default Report Server URL**

**Syntax:** - HTTP://<IP ADDRESS/SERVERNAME>:PORT/REPORT SERVER NAME

**Ex:** - http://vinaytech-pc:80/ReportServer-8AM-batch

### **Changing:**

**Virtual Directory : RS\_Vinay**

**Data Bases:** This is the place where we can monitor and change the existing reporting service database & credentials..

### **Default Data Bases**

Report Server DB , Report ServerTemp DB

### **Changing Database**

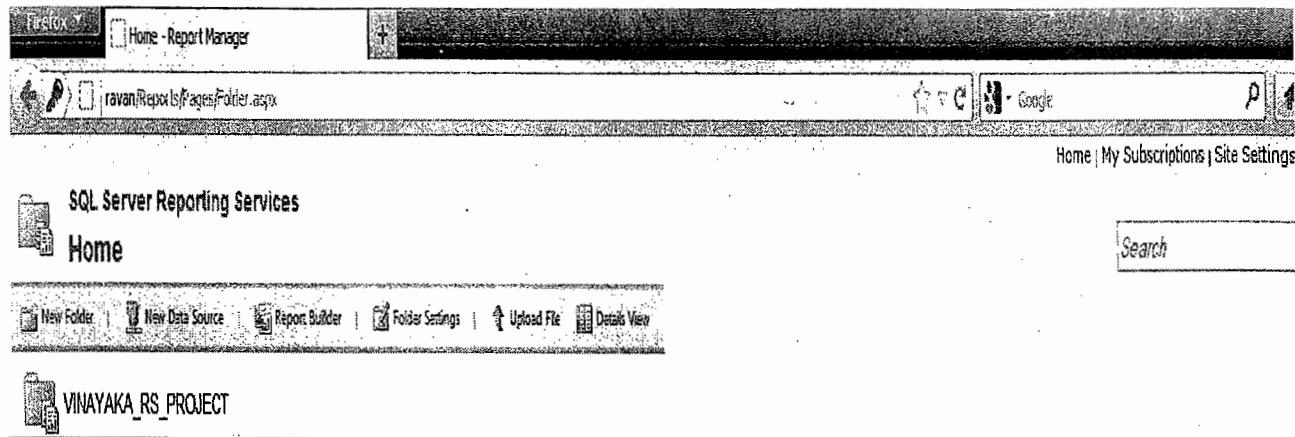
Click change Database name → Specify server name, database name and click finish.

If you want see databases

### **Navigation:**

### **Report Manager URL:**

Settings for report manager available.



### **Default report manager:**

Syntax:

HTTP://<IP ADDRESS(or) SERVER NAME>:PORT <REPORT MANAGER NAME>

Ex: http://vinaytech-pc:80/Reports-Batch

Changing to

**Execution Account:** - Specify the account to enable usage of report data sources that get credentials or to connect remote servers that stored in external images used in reports.

**Email Setting:** - It takes SMTP server send the address. We should take proper email settings to perform subscription operations, other scheduling operations easily.

**Encryption Keys:** Backup and restore encryption key

The keys are saved with an extension called ".SNK". Here real time use MD5 algorithm.

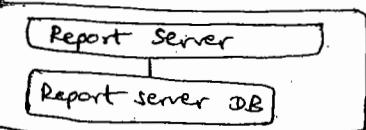
### **Scaleout Deployment**

Deployment

#### **single server Deployment**

Report server and DB in the same machine.

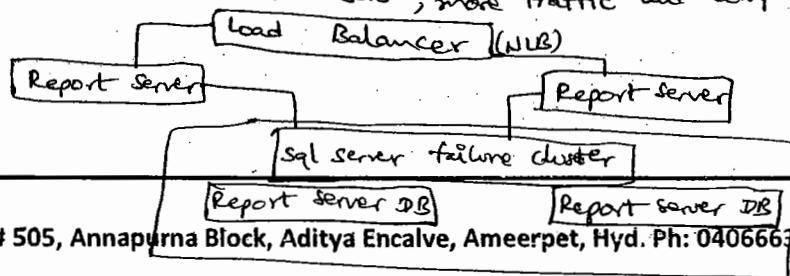
Recommended for less users, less traffic, simple and less reports.



#### **Scaleout deployment**

Report server and DB in diff. machines

Recommended for more users, more traffic and complex & many reports.



## \*\* CREATING REPORTS \*\*

There are ~~three~~<sup>four</sup> ways to create a report

- By using Report Server project wizard
- Report Server project
- Report Builder

d) Power view

## \*\* CLIENT APPLICATIONS \*\*

Tools required to operate Reporting Services

SSRS includes several client applications that use the SSRS programming interface, including Web service APIs and URL access methods, to provide front-end tools for users to access both SSRS reports and configuration tools. These tools provide report server management, security implementation, and report-rendering functionality. The tools are as follows:

- Report Manager:** This browser-based application ships with SSRS and provides a graphical interface for users who need to view or print reports, or to manage report objects for their workgroups or departments.
- BIDS or SSDT:** This tool provides an integrated environment for developing SSRS reports.
- Command-line utilities:** You can use several command-line tools to configure and manage the SSRS environment, including rs, rsconfig, rskeymgmt and rsactivate.
- Report Builder 3.0:** This enhanced application was primarily developed to give business users the ability to create ad hoc reports. Nearly all of the features available in BIDS are also available in Report Builder 3.0.
- Custom clients:** These .NET Windows Forms and Web applications call the SSRS Web service to perform such tasks as rendering reports and managing report objects. SSRS includes sample application projects that you can compile and run to extend the functionality provided by the main tools listed earlier.

## \*\* COMMANDLINE UTILITIES \*\*

To work from Command line the below Commands are helpful.

In addition to graphical applications such as BIDS and SSMS, SSRS provides several command-line utilities that are considered Web service clients. The tools have the added benefit of being automated by using built-in task scheduling in Windows. SSRS includes four main command-line utilities:

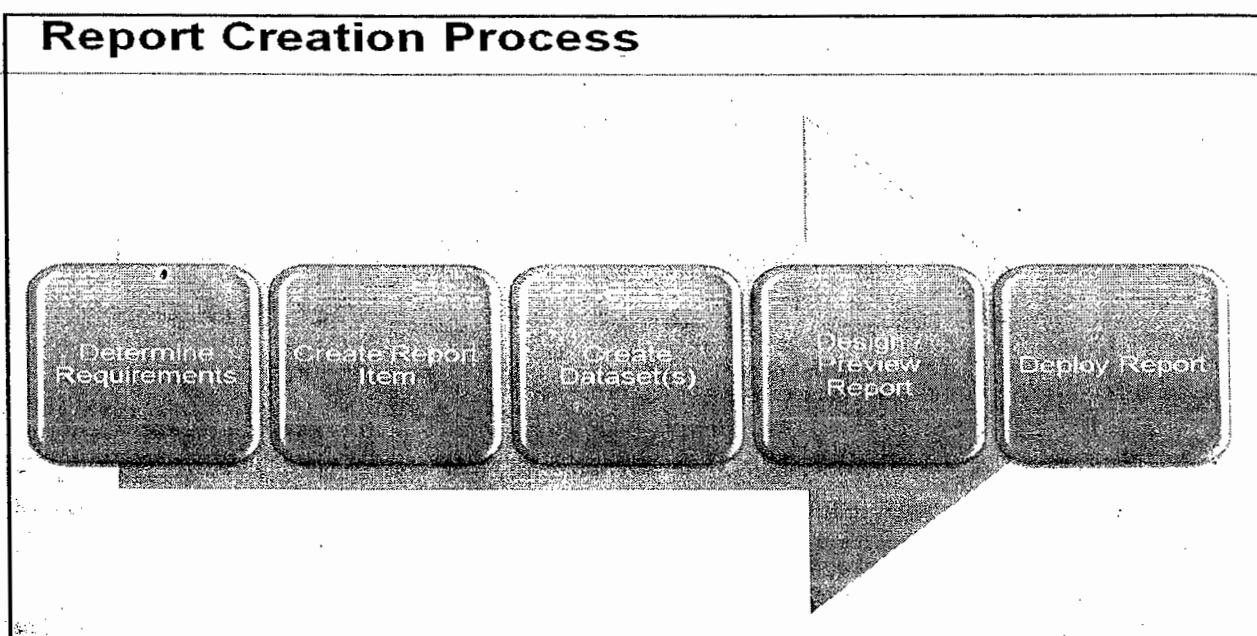
- rs: Processes reporting services script (RSS) files that are written with VB.NET code. Because you can access the full SSRS API via the code in the script, all SSRS Web service methods are available.
- rsconfig: Configures the authentication methods and credentials for connecting SSRS to the ReportServer database. The rsconfig utility also sets the unattended SSRS execution credentials.
- rskeymgmt: Manages the encryption keys that SSRS uses to store sensitive data securely, such as authentication credentials.
- rsactivate: Adds another instance of Reporting Services to a Web Farm, and is useful for replacing a corrupted instance.

## Custom Clients

The final types of clients are those custom designed to access the SSRS Web services, which use both the report viewer and the report publisher. Third-party commercial applications provide extended functionality, for instance RS Scripter, which assists in scripting and managing various report server catalog items; it can extract RDLS from a server, and manage subscriptions, roles, and even migration from one server to another.

### **\*\* STEPS TO CREATE A REPORT \*\***

AND DELIVER :



**STEPS:** Customer Requirement: Every day the report should be delivered PDF format at morning 9 AM to an email-id [vinaytech@gmail.com](mailto:vinaytech@gmail.com).

STEP 1: Understand Business Requirements, Read Report Def. documents.

STEP 2: Open SSDT/ any other relevant tool and implement the below steps

a) Take layout and concentrate on design

b) Create data source (connection to a DB or file)

c) Taking req. columns from the data source as a dataset.

Note: <sup>look at</sup> To get data from DB query (or) procedure (or) view helpful.

d) Place dataset columns on design

e) preview the report

STEP 3: To publish <sup>go to</sup> project Project properties and specify target server URL as report server URL.

STEP 4: Build Deploy.

**Dataset:** - It is a logical object for physical collection of tables, it takes the required columns from the data source.

**Generally it takes**

- a. Query
  - c. View
  - b. Procedure
  - d. Query from a file

## **\*\* DATA REGIONS \*\***

Region : Region holds multiple values.

1. Table: For Columnar reporting this is helpful.
  2. Matrix: For cross tab reporting this is helpful.
  3. chart: To present numerical vs textual in graph format.
  4. Gauge: To point a value in a range.
  5. Indicator: To present important business Indicators such as KPI (Key Performance Indicator)
  6. Map: To generate maps from spatial coordinates.
  7. Data bar, Sparkline: Category wise (or) range wise value change (or) value distribution it shows.
  8. List: For free form Reporting.
  9. Sub Report: Report Insert another Report.

## **REPORT ITEMS**

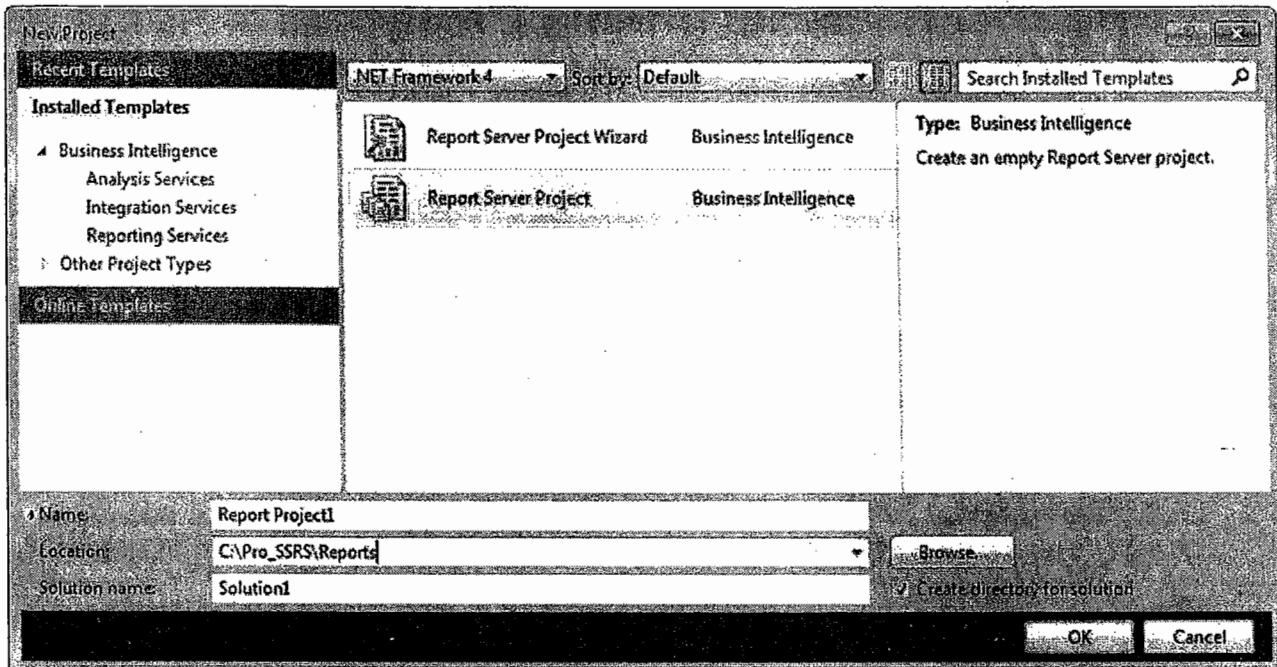
Item holds single value.

1. Text box: Displays a value based on expression.
  2. Line: Helps as to draw a line.
  3. Pointer: For taking a pointer information.
  4. Image: Displaying image from external source (or) database etc.

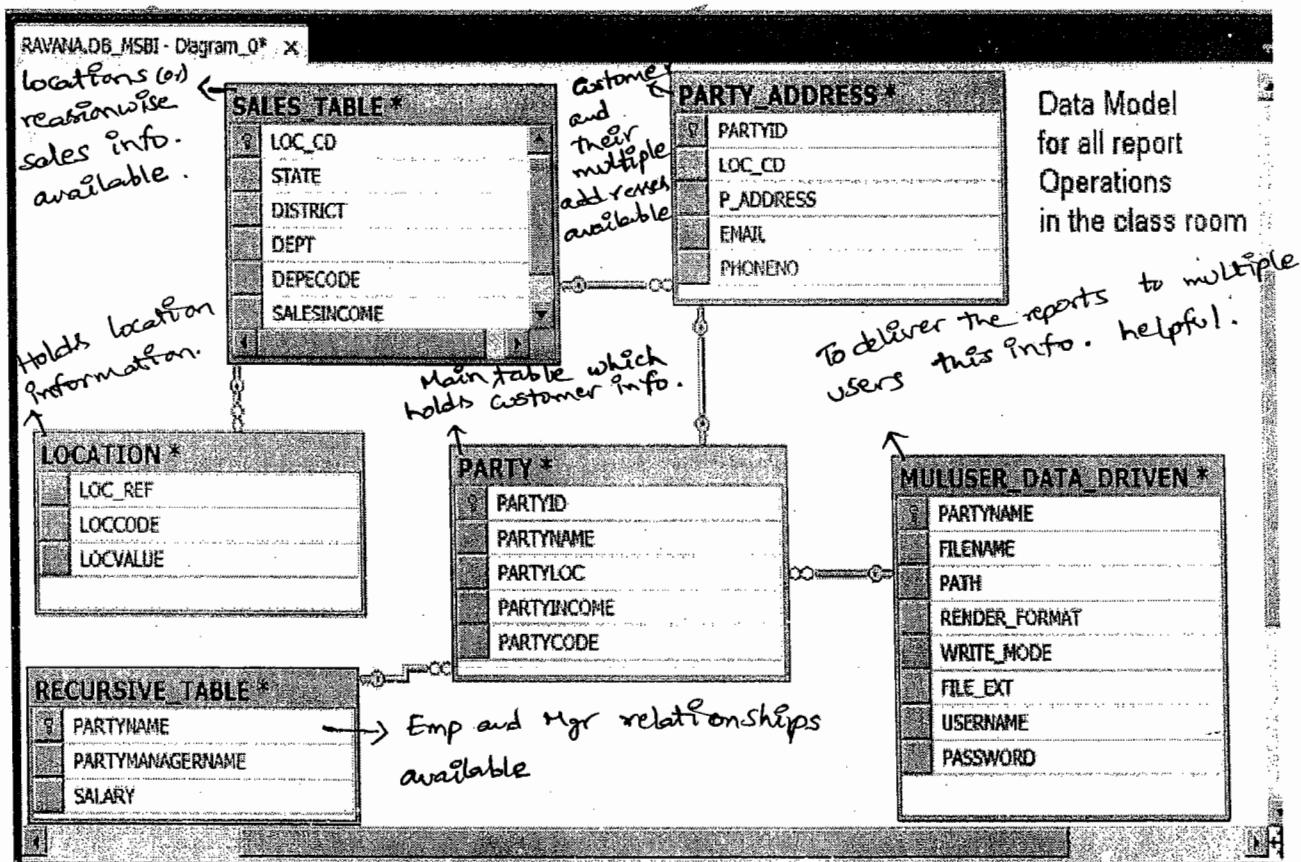
Shared Data Source: This data source can be used across multiple reports in a project.

Shared Data set: It can be used across multiple reports in a project.

## WINDOW FOR REPORT OPERATIONS



## TABLES FOR REPORT OPERATIONS & DATA MODEL



Note: All these tables available under DB\_MSBI in soft copies.

## CREATING A REPORT THROUGH SERVER WIZARD

In this approach we have self driven steps with less manual interaction.

1. Open ~~SSDT~~ SSDT, file → new project choose template as reporting services, specify project name, solution name, choose template as report server project wizard.

Next →

Specify Data Source settings by clicking Edit →

Next

Write a Query / Click Query Builder for constructing a query

(Ex: Select \* RFrom Party)

Next

Select Report Item (Table /Matrix)

Next

Specify the available fields into Displayable fields according to the design

Ex: PartyLoc into Page

PartyCode into Group

Partyid, Partyname into Detail

Next

Specify Stepped /Blocked and Subtotals (if required)

Next

Specify Style

(Bold)

Next

Specify ReportServer name :

ex: http://ravana:8080/reportserver (To know this goto report manager tool)

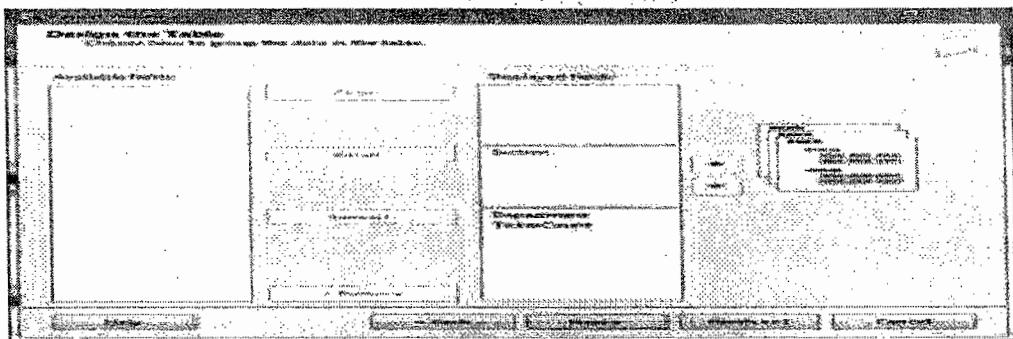
Next → Finish

Preview (Verify look ,Feel and Content)

Build->Deploy

Goto ReportServer (Developer Friendly GUI) or Report Manager (Customer Friendly GUI) to see the report!

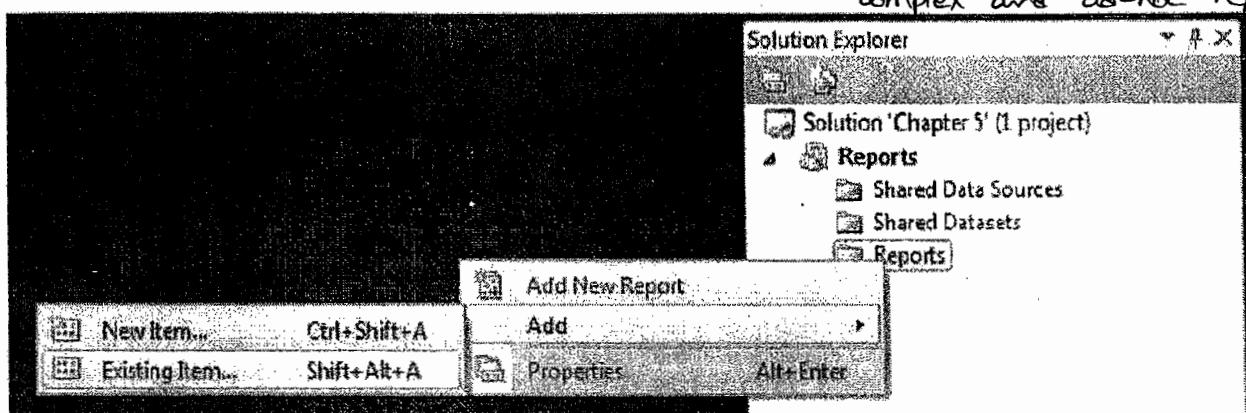
Create a subscription which delivers through e-mail.



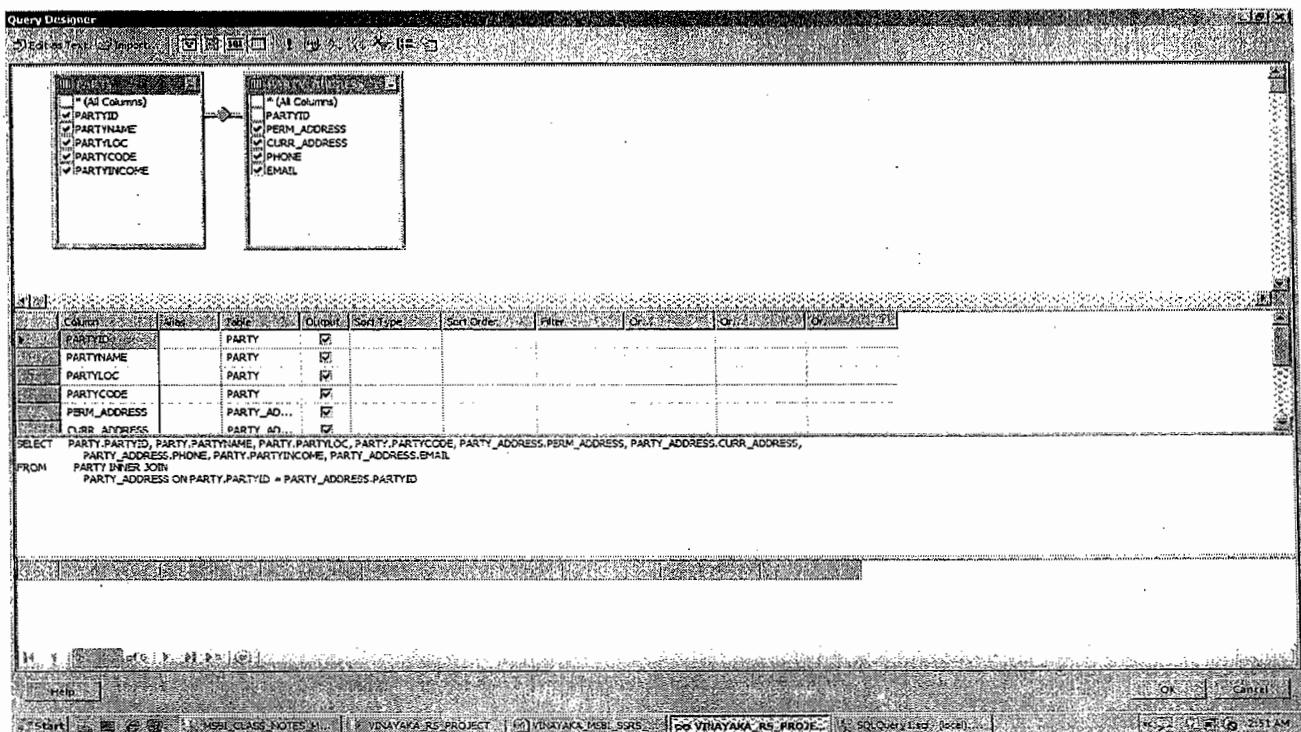
Note : To implement this goto Report Manager and subscription mechanism.

## \* CREATING A STANDARD REPORT

This standard method only we follow in real time for creating simple and ad-hoc reporting.



1. Open SSDT, file → New project - specify project name, folder and solution name - next (choose reporting services - Report server project).
2. View - solution explorer - reports - right click (rc) - add new item, choose report and specify the report name as Party - party address.
3. View - report data - create a data source, dataset like below,
  - a) data source creation → rc → new data source - specify server type as sql server, click edit - specify server name, db name - ok.
  - b) data set → rc - new dataset - specify a query (or) call a procedure (or) import sql - ok.
4. View - tool box - drag and drop table into the design
5. By choosing dataset columns (or) by dragging and dropping dataset columns fill the table cell values. - preview.
6. View - solution explorer - project - rc - properties - specify target server URL as Report Server URL.
7. Build Deploy.
8. Use push method to subscribe and deliver (or) pull method to directly access.



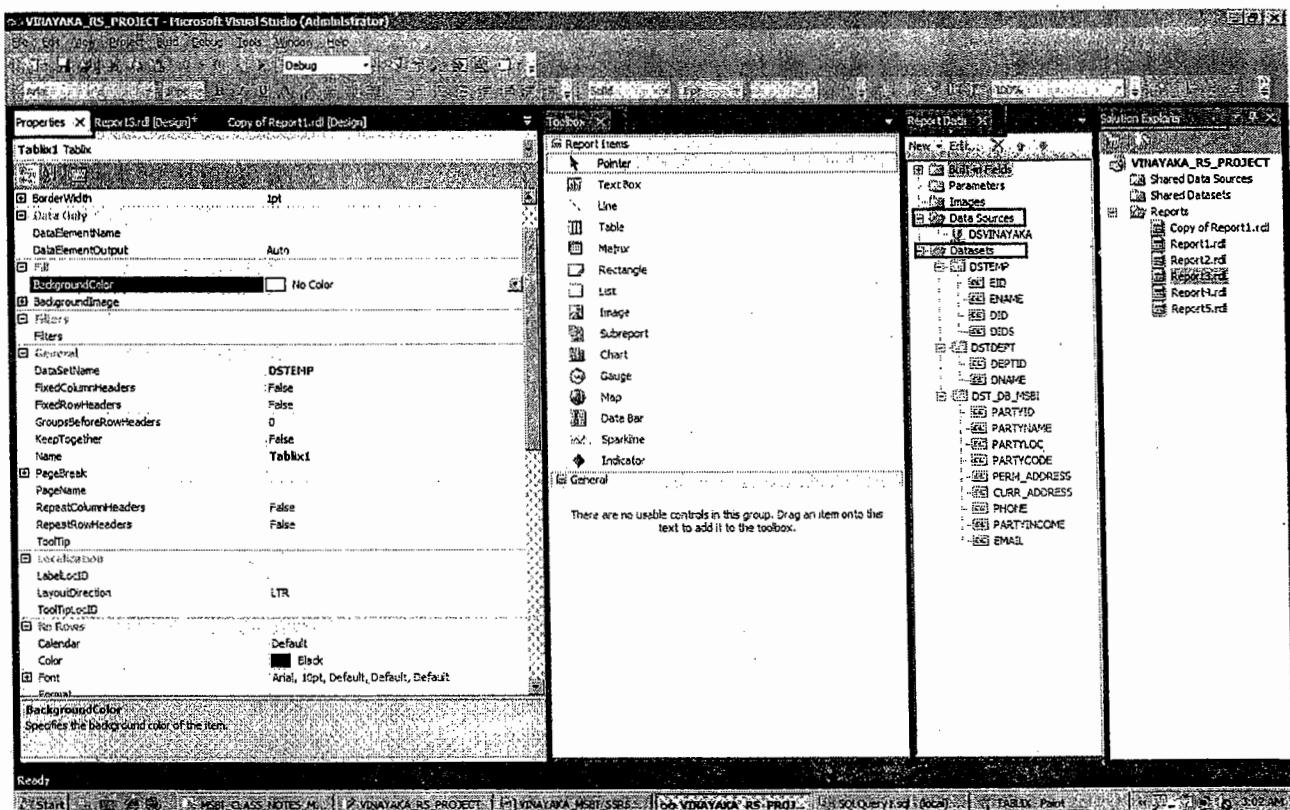
### REAL TIME INPUT & OUTPUT DOCUMENTS

Refer to first page as RS.

# Vinay Tech House

### IMPORTANT NAVIGATIONS

- View - Solution Explorer: Describes about solution, project, Reports.
- View - Tool Box: Displays Tool box items such as regions and normal items.
- View - Report Data: Shows individual report settings such as datasource, dataset, parameters etc.
- View - Output: Shows the statistics of the report.
- View - Error List: Displays the list of errors.



## ALL OBJECTS PROPERTIES AND FORMATIONS

**They are two ways**

- Highlight the object (or) section - F4 / F8 - properties / view - properties .
- Go to format menu and use various format options.

Exercise: Try to observe changing background, foreground colors, border length and style and font style with etc. with left alignment (or) center alignments.

### Assign Borders, width, Styles, etc. . . . :-

Select the table press F4

Border Color : Black

Border Style : Solid

Border Width : 2pt

Go to the corresponding section press F4 and change the border color, style and width

*connect*

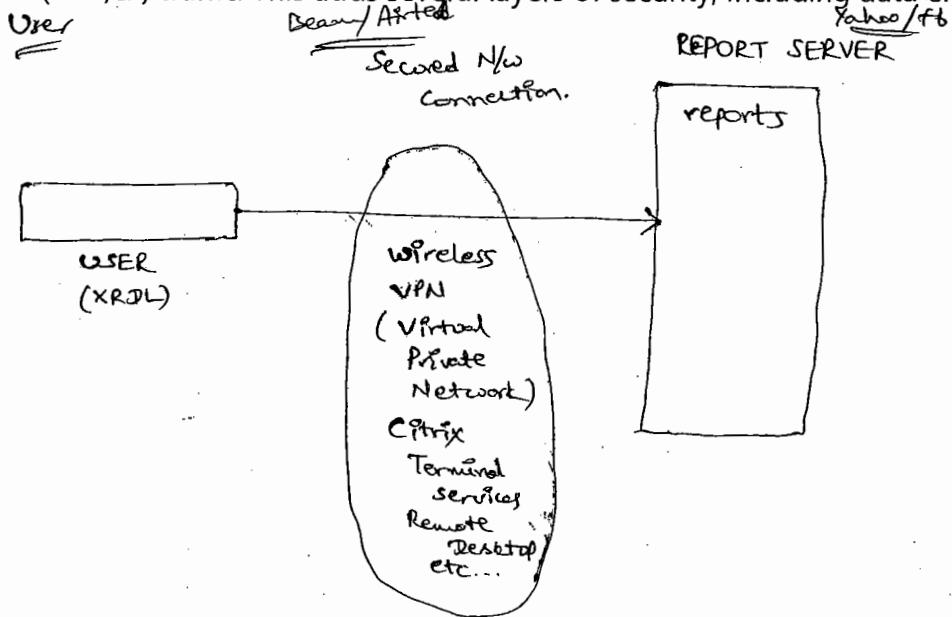
## ENCRYPTING DATA

How does SSRS client & SSRS server (or) what are the imp. components required to connect to server.

When working with confidential data of any kind, the chief concern is that the only people who can see the data are those who need to see it and who have been specifically granted permission to see it. This is especially true of Protected Health Information (PHI) data, as defined by HIPAA, which has been a significant concern of ours as a software development company. Many other types of data also need this level of protection, including financial, HR, and many more. We'll start with the first of the three main challenges we defined as crucial to a successful, secure deployment of SSRS; namely, data encryption.

**Introducing Encryption** In today's mixed-technology networked environment, data encryption comes in many varieties. However, regardless of the technology, the encryption algorithms must meet a high standard for complexity and reliability. Fortunately, many applications provide built-in levels of encryption. SSRS natively supports encrypting the sensitive data it stores in the ReportServer database and configuration files. Companies may have the following other technologies in place that can be used in conjunction with SSRS encryption:

- **Wireless:** Uses Wireless Encryption Protocol (WEP), with shared keys to encrypt data transmitted through wireless access points.
- **HTTPS:** Uses a server certificate, generally from a trusted authority such as VeriSign, to provide encryption over Secure Sockets Layer (SSL). SSL is used when transmitting data with HTTPS instead of HTTP.
- **Terminal Services:** Uses Remote Desktop Protocol (RDP) for connecting remotely from a client workstation to a terminal server. This provides four levels of data encryption in Windows: Low, Client Compatible, High, and FIPS Compliant.
- **VPNs:** Allows accessibility to internal networks from VPN client systems. Encapsulates and encrypts Point-to-Point Tunneling Protocol (PPTP) and Layer 2 Tunneling Protocol (L2TP).
- **IPSec:** Is the standard security protocol for Transmission Control Protocol/Internet Protocol (TCP/IP) traffic. This adds several layers of security, including data encryption.



Before report connected to report server it always processed through secured n/w area

first

## DEPLOYING REPORTS

SSRS 2012 provides several means for deploying reports:

### **Using the Report Manager interface through your Web browser:**

This simple method allows anyone with an RDL file and the proper SSRS permissions to upload it to the SSRS 2012 server. This can be especially useful if you're developing your report's RDL files in an application that doesn't provide you with a method to upload them to the server. It's also useful if you want to make a quick edit of the RDL file—say to change a misspelled word using an application such as Notepad, which doesn't offer a built-in way to upload the report.

### **Using Report Builder 3.0:**

The Report Builder 3.0 utility was a new feature to SSRS 2008 R2 and it is still used in SSRS 2012. It provides the user who is unfamiliar with the Visual Studio IDE with a simple interface by which to create and edit data sources and reports.

### **Using the Deploy option in BIDS/Visual Studio:**

This method allows you to deploy your reports to the SSRS 2012 server from directly within your development environment. If you're using BIDS/Visual Studio and have direct access to the report server to which you want to deploy your reports, this is one of the easiest options.

### **Using the rs.exe command-line utility:**

The rs.exe command-line utility is a runtime environment that is used to execute VB .NET code in the form of specially formatted script files. You deploy the report in the same way as the method in the next item of this list. Later in the section "Using the rs.exe Utility," we will cover how to deploy reports and create data sources programmatically using rs.exe.

### **Programmatically, using the SSRS Webservice:**

This method gives you complete control over the deployment process with the added advantage of creating any type of UI you want. Unlike the rs command-line utility, you have your choice of languages and the full power of Visual Studio 2012 to help you develop your custom interface. The SOAP API, otherwise known as the Report Server Web service, to retrieve report parameter information about reports from the SSRS 2012 server, and then used that information to generate a Windows Forms UI for parameter selection.

## WORKING WITH TABLE DATA REGION

It contains the below sections

a) **Header Section:** - Here columns headings or aggregated information displayed.

b) **Data Section:** - Here data for the columns display.

c) **Footer Section:** columns headings/footings or aggregated information displayed.

c) **Group Section:** -

1. It comes under additional section to have groups of data or divide into multiple views.
2. This group section comprises header, footer (summarized sections) sections.
3. Take Party Id, Party Name etc. . . sections so that headings will come in the header section we can change the heading as we required.

### A. Adding columns to left or right.

1. Go to the table → right click → insert column → left (or) right
2. Select the fields from dataset drag and drop into table region.

### WORKING WITH TABLIX PROPERTIES

These properties are diff. from general properties. These define only data changes where as general properties deal with structure.

These properties are common for Table, Matrix and List. It contains four sections with different options.

a) General Section      Click on general → tooltip → This is product info.

**Vinay Tech House**

Name : FIX PARTY  
Tool Tip : PARTY PARTY ADDRESS DATA

Dataset Name : DST\_MSBI

Select the table goto the left most corner → rc → click on tablix prop.

\* c) Visibility click on visibility → select the option show / Hide based on expression.

- i. Show - Displays the object      Click on expression button,
- ii. Hide - Hides the object      =if(sum(fields!salesAmount.Value,"dataset1") > 500, false, true)
- iii. Show or hide base on the expression

Text Info - Select the text box press F4. Observe the textbox name.  
Tablix prop → visibility → tick → check Textbox24  
iv. Display can be toggled by the report item. Click on OK → observe preview.

1. To implement drilldown this process is useful.

2. Display the report when we expand the textbox (which is available separately).

Navigation: Drag & drop textbox from report item. Provide the text within the textbox.

d) Filters: - 1. Filter the data before it displays. Filters → click on Add button

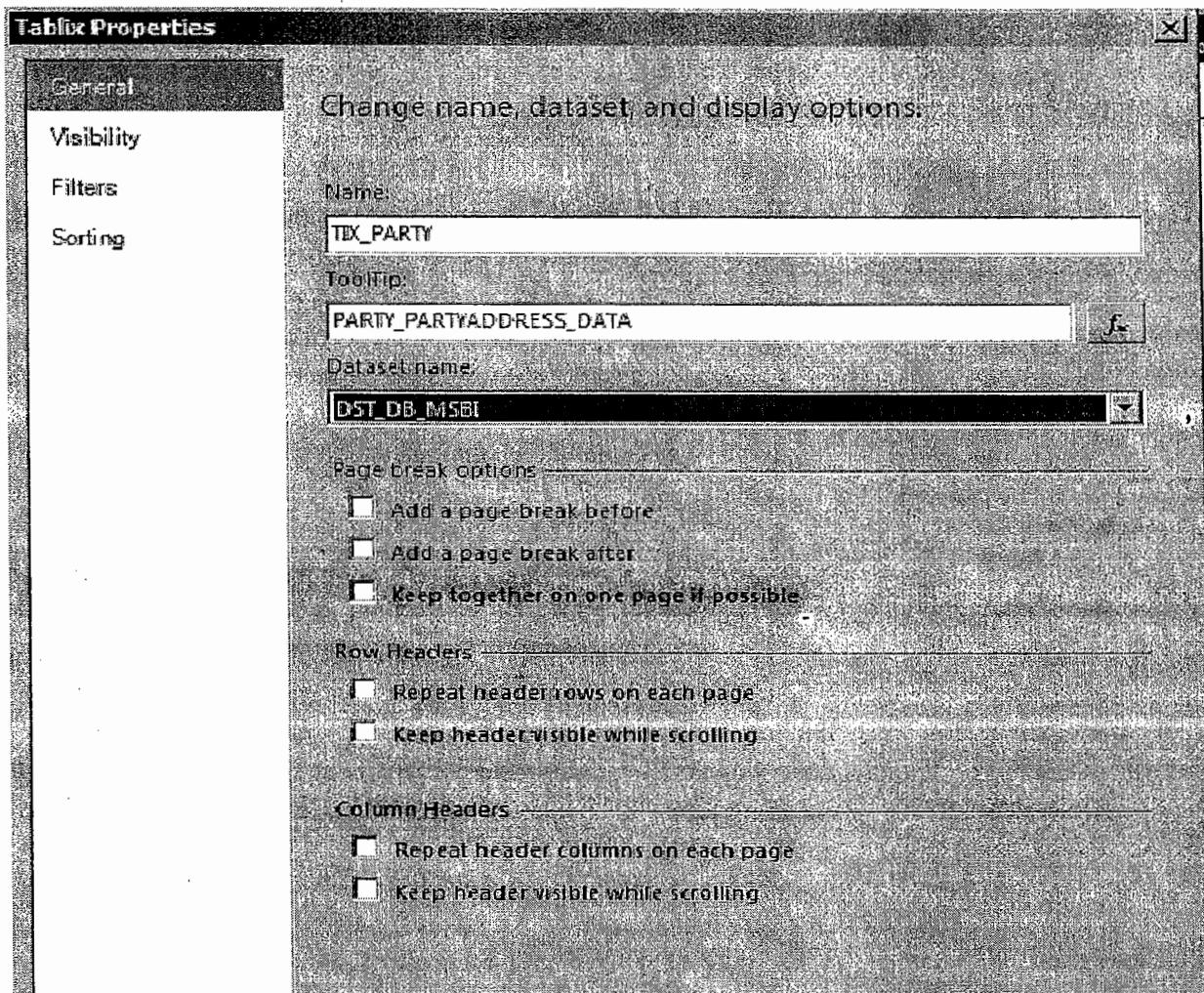
Expression : salesamount  
operator : >  
value : ✓

\* Create proc p1  
as  
begin  
— SELECT STATEMENT  
end

e) Sorting: - It displays the data in the sorting order (ascending / descending).

A-Z → Ascending Order Z-A → Descending Order

Tablix prop → Sorting.



~~Apply operations expressions formations etc this concept is required.~~

## MSBI 2012 (SQL Server Integration Services)

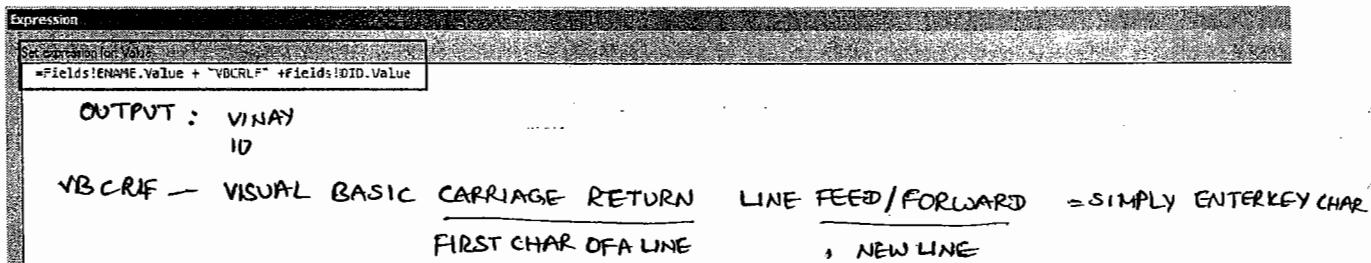
\* Interview oriented not required but to perform any intermediate operations very good practice is mandatory.

## TEXTBOX EXPRESSIONS & PROPERTIES

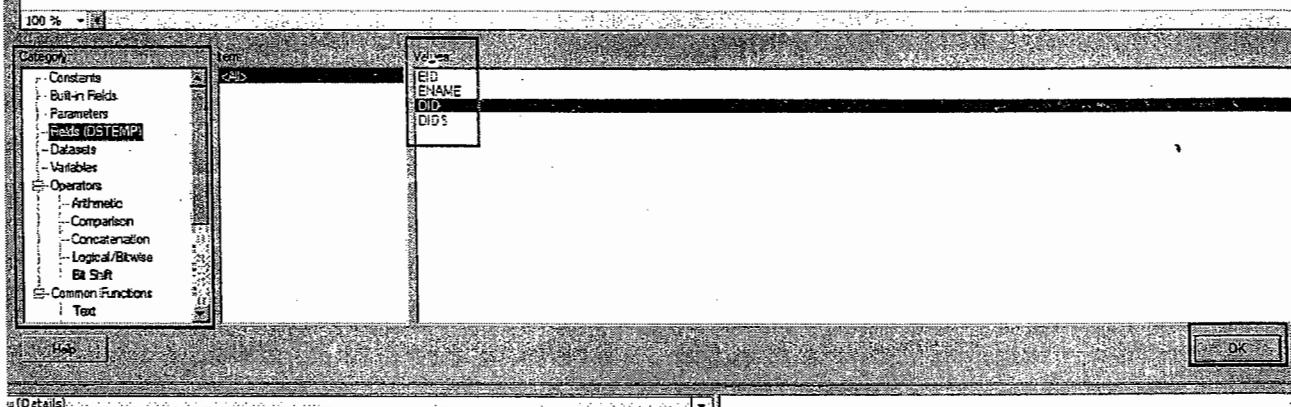
**Expression:** - Here we can write an expression by using the below components.

### Expression Examples in Reporting Services

Some expressions are frequently used in reports. These include expressions to change the appearance of data in a report, change properties of report items, and affect how data is retrieved. This topic describes some expressions that can be used for common tasks in a report.



Always text box (or) a cell displays expression result only. (May be directed value or operated value).



## Functions

Many expressions in a report contain functions. You can format data, apply logic, and access report metadata using these functions. You can write expressions that use functions from the Visual Basic Run-Time Library, and from the System.Convert and System.Math namespaces. You can add references to functions from other assemblies or custom code. You can also use classes from the Microsoft .NET Framework, including System.Text.RegularExpressions. For more information about Visual Basic functions supported in expressions, see "Visual Basic Run-Time Library" at [msdn.microsoft.com](http://msdn.microsoft.com).

## Visual Basic Functions

You can use Microsoft Visual Basic functions to manipulate the data that is displayed in text boxes or that is used for parameters, properties, or other areas of the report. This section provides examples demonstrating some of these functions. For more information about Visual Basic functions, see the Visual Basic documentation.

## Date Functions

You can use Visual Basic functions to provide date information in your report.

- The following expression contains the **Today** function, which provides the current date. This expression can be used in a text box to display the date on the report, or in a parameter to filter data based on the current date.

**=Today()**

- The **DateAdd** function is useful for supplying a range of dates based on a single parameter. The following expression provides a date that is six months after the date from a parameter named **StartDate**.

**=DateAdd(DateInterval.Month, 6, Parameters!StartDate.Value)**

- The following expression contains the **Year** function, which displays the year for a particular date. You can use this to group dates together or to display the year as a label for a set of dates. This expression provides the year for a given group of order dates. The **Month** function and other functions can also be used to manipulate dates. For more information, see the Visual Basic .NET documentation.

## Vinay Tech House

### String Functions

You can use Visual Basic functions to manipulate strings in your report.

- You can combine more than one field by using concatenation operators and Visual Basic constants. The following expression returns two fields, each on a separate line in the same text box.

**=Fields!FirstName.Value & vbCrLf & Fields!LastName.Value**

- You can format dates and numbers in a string with the **Format** function. The following expression displays values of the **StartDate** and **EndDate** parameters in long date format.

**=Format(Parameters!StartDate.Value, "D") & " through " & Format(Parameters!EndDate.Value, "D")**

If the text box contains only a date or number, you should use the **Format** property of the text box to apply formatting instead of the **Format** function within the text box.

- The **Right**, **Len**, and **InStr** functions are useful for returning a substring, for example, trimming `DOMAIN\username` to just the user name. The following expression returns the part of the string to the right of a backslash (\) character from a parameter named **User**:

```
=Right(Parameters!User.Value, Len(Parameters!User.Value) - InStr(Parameters!User.Value, "\"))
```

The following expression results in the same value as the previous one, using members of the .NET Framework **String** class instead of Visual Basic

```
=Parameters!User.Value.Substring(Parameters!User.Value.IndexOf("\") + 1, parameters!User.Value.Length - Parameters!User.Value.IndexOf("\") - 1)
```

- You can display the selected values from a multivalue parameter. The following example uses the **Join** function to concatenate the selected values of the parameter **MySelection** into a single string that can be set as an expression for the value of a text box in a report item.

```
= Join(Parameters!MySelection.Value)
```

If the text box contains only a date or number, you should use the **Format** property of the text box to apply formatting instead of the **Format** function in the text box.

- The **Regex** functions from the .NET Framework **System.Text.RegularExpressions** are useful for changing the format of existing strings, for example, formatting a telephone number. The following expression uses the **Replace** function to change the format of a ten-digit telephone number in a field from the form `nnn-nnn-nnnn` to the form `(nnn) nnn-nnnn`:

```
=System.Text.RegularExpressions.Regex.Replace(Fields!Phone.Value, "(\d{3})[ -.]*(\d{3})[ -.]*(\d{4})", "($1) $2-$3")
```

### **Conversion Functions**

You can use Visual Basic functions to convert data types as needed in your report.

- You can use Visual Basic functions to convert data types as needed. Convert functions are frequently used to eliminate or format #Error messages in a report. The following expression displays the number of values selected for the multivalue parameter **MySelection**.

```
=CStr(Parameters!MySelection.Count)
```

## **Decision Functions**

You can use Visual Basic functions to evaluate an input value and return another value depending on the result.

- The **Iif** function returns one of two values depending on whether the expression evaluated is true or not. The following expression uses the **Iif** function to return a Boolean value of True if the value of **LineTotal** exceeds 100. Otherwise it returns False:

```
=Iif(Fields!LineTotal.Value > 100, True, False)
```

The following expression uses multiple **Iif** functions (also known as "nested Iifs") to return one of three values depending on the value of **PctComplete**.

```
=Iif(Fields!PctComplete.Value >= .8, "Green", Iif(Fields!PctComplete.Value >= .5, "Amber", "Red"))
```

- The following expression also returns one of three values based on the value of **PctComplete**, but uses the **Switch** function instead, which returns the value associated with the first expression in a series that evaluates to true:

```
=Switch(Fields!PctComplete.Value >= .8, "Green", Fields!PctComplete.Value >= .5, "Amber", Fields!PctComplete.Value < .5, "Red")
```

- The following expression tests the value of the **ImportantDate** field and returns "Red" if it is more than a week old, and "Blue" otherwise. This expression can be used to control the Color property of a text box in a report item.

```
=IIF(DateDiff("d", Fields!ImportantDate.Value, Now())>7, "Red", "Blue")
```

- The following expression tests the value of a field **PhoneNumber** to see whether it is **null** (**Nothing** in Visual Basic) and returns "No Value" instead of the null value. This expression can be used to control the value of a text box in a report item.

```
=IIF(Fields!PhoneNumber.Value Is Nothing, "No Value", Fields!PhoneNumber.Value)
```

- The following expression tests the value of the **Department** field and returns either a subreport name or a **null** (**Nothing** in Visual Basic). This expression can be used for conditional drillthrough subreports.

```
=Iif(Fields!Department.Value = "Development", "EmployeeReport", Nothing)
```

- The following expression tests if a field value is null. This expression can be used to control the Hidden property of an image report item.

```
=IIf(IsNothing(Fields!LargePhoto.Value), True, False)
```

## Report Functions

Reporting Services provides additional report functions that you can use to manipulate data in a report. This section provides examples for two of these functions. For more information about report functions and examples,

The **Sum** function can total the values in a grouping or data region. This function can be useful in the header or footer of a table group. The following expression displays the sum of data in the Order grouping or data region:

```
=Sum(Fields!LineTotal.Value, "Order")
```

- An expression containing the **RowNumber** function, when used in a text box within a data region, displays the row number for each instance of the text box in which the expression appears. This function can be useful to number rows in a table. It can also be useful for more complex tasks, such as providing page breaks based on number of rows. For more information, see "Page Breaks" later in this topic.

The following expression displays the row number from the first row in the outermost data region to the last. The **Nothing** keyword indicates that the function will start counting at the first row in the outermost data region. To start counting within child data regions, use the name of the data region.

```
=RowNumber(Nothing)
```

## Appearance of Report Data

You can use expressions to manipulate how data appears on a report. For example, you can display the values of two fields in a single text box, display information about the report, or affect how page breaks are inserted in the report.

## Page Headers and Footers

When designing a report, you may want to display the name of the report and page number in the report footer. To do this, you can use the following expressions:

- The following expression provides the name of the report and the time it was run. It can be placed in a text box in the report footer or in the body of the report. The time is formatted with the .NET Framework formatting string for short date:

```
=Globals.ReportName & ", dated " & Format(Globals.ExecutionTime, "d")
```

- The following expression, placed in a text box in the footer of a report, provides page number and total pages in the report:

**=Globals.PageNumber & " of " & Globals.TotalPages**

You can also use expressions in the report header or footer to report items from the body of the report. The following examples describe how to display the first and last values from a page in the page header, similar to what you might find in a directory listing. The example assumes a data region that contains a text box named **LastName**.

- The following expression, placed in a text box on the left side of the page header, provides the first value of the **LastName** text box on the page:

**=First(ReportItems("LastName").Value)**

- The following expression, placed in a text box on the right side of the page header, provides the last value of the **LastName** text box on the page:

**=Last(ReportItems("LastName").Value)**

You can apply aggregates to a report item reference in a page header or footer. (However, you cannot apply an aggregate to a report item reference in the report body.) The following example describes how to display a page total. The example assumes a data region that contains a text box named **Cost**.

- The following expression, placed in the page header or footer, provides the sum of the values in the **Cost** text box for the page:

**=Sum(ReportItems("Cost").Value)**

**Note:** You can refer to only one report item per expression in a page header or footer.

## Page Breaks

In some reports, you may want to place a page break at the end of a specified number of rows instead of, or in addition to, on groups or report items. To do this, create a group in a data region (typically a group immediately outside the detail), add a page break to the group, and then add a group expression to group by a specified number of rows.

- The following expression, when placed in the group expression, assigns a number to each set of 25 rows. When a page break is defined for the group, this expression results in a page break every 25 rows.

**=Int((RowNumber(Nothing)-1)/25)**

## Properties

Expressions are not only used to display data in text boxes. They can also be used to change how properties are applied to report items. You can change style information for a report item, or change its visibility.

## Formatting

You can use expressions to vary the appearance of report items in a report.

- The following expression, when used in the **Color** property of a text box, changes the color of the text depending on the value of the **Profit** field:

`=Iif(Fields!Profit.Value < 0, "Red", "Black")`

- The following expression, when used in the **BackgroundColor** property of a report item in a data region, alternates the background color of each row between pale green and white:

`=Iif(RowNumber(Nothing) Mod 2, "PaleGreen", "White")`

If you are using an expression for a specified scope, you may have to indicate the dataset for the aggregate function:

`=Iif(RowNumber("Employees") Mod 2, "PaleGreen", "White")`

## Visibility

You can show and hide items in a report using the visibility properties for the report item. In a data region such as a table, you can initially hide detail rows based on the value in an expression.

- The following expression, when used for initial visibility of detail rows in a group, shows the detail rows for all sales exceeding 90 percent in the **PctQuota** field:

`=Iif(Fields!PctQuota.Value>.9, False, True)`

- The following expression, when set in the **Visibility, Hidden** property of a table, shows the table only if it has more than 12 rows:

`=IIF(CountRows()>12,true,false)`

## Report Data

Expressions can be used to manipulate the data that is used in the report. You can refer to parameters and other report information. You can even change the query that is used to retrieve data for the report.

## Parameters

You can use expressions in a parameter to vary the default value for the parameter. For example, you could use a parameter to filter data to a particular user based on the user ID that is used to run the report.

- The following expression, when used as the default value for a parameter, collects the user ID of the person running the report:

=User!UserID

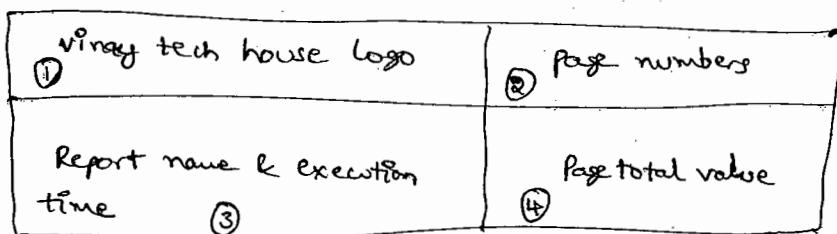
- You can use the following expression to refer to the parameter in a query parameter, filter expression, text box, or other areas of the report. This example assumes that the parameter is named **User**:

=Parameters!User.Value

## WORKING WITH PAGE HEADER AND FOOTER

- Usually built-in fields info. is helpful to display the content on page header and footer
- These fields can also be called as global variables.
- Generally these variables called page no's, report name, execution time etc.
- The greatest advantage with page header and footer is to display company logos, report aggregate information and page statistics.

- Design:
- Report Menu → add page header
  - Report Menu → add page footer



1 → Take an image → specify embedded option and browse the image location.  
Goto size section and select fit to size option.

2 → Globals! page no's & "-To" & Globals! Total pages

## CUSTOM CODE WRITING, ASSEMBLIES AND USAGE

The expected functionality you don't find in IS, AB, RS level we use Scripting and programming to implement that. That coding is called Custom Coding.

Both BI Components and custom Coding run under CLR. (Good part).

There are two types of Coding. a) VB Script in ReportLevel to use anywhere in the Script.

b) .NET programming to use across all reports by keeping in a assembly level.

**Point1:** The following example calls the embedded code method **ToUSD**, which converts

to indicate custom operation the **StandardCost** field value to a dollar value:

=Code.ToUSD(Fields!StandardCost.Value)

**Point2:** The following example shows how to define some custom constants and variables.

[Visual Basic]

```
Public Const MyNote = "Authored by Bob"
Public Const NCopies As Int32 = 2
Public Dim MyVersion As String = "123.456"
Public Dim MyDoubleVersion As Double = 123.456
```

Although custom constants and variables do not appear in the Expression Editor

**Point3:** Constants view (which only displays built-in constants), you can add references to them from any expression, as shown in the following examples. These are treated as Variants.

[Visual Basic]

```
=Code.MyNote
=Code.NCopies
=Code.MyVersion
=Code.MyDoubleVersion
```

} Variables usage

**Example 1: (Step1)** The following example calls the embedded code method **FixSpelling**, which substitutes Bicyclefor all occurrences of the text Bike in SubCategory.Value.

=Code.FixSpelling(Fields!SubCategory.Value)

**(Step2)** The following code, when embedded in a report definition, shows an implementation of the **FixSpelling** method. The first time this custom code runs, a **MessageBox** displays the

substituted text. This example shows you how to refer to the .NET Framework **StringBuilder** class and the **System.Windows.Forms.MessageBox** class. You must add a reference to your report properties for **System.Windows.Forms**

```

Dim firstTime As Boolean = True
Public Function FixSpelling(ByVal s As String) As String
    Dim strBuilder As New System.Text.StringBuilder(s)
    If s.Contains("Bike") Then
        strBuilder.Replace("Bike", "Bicycle")
    If firstTime Then
        System.Windows.Forms.MessageBox.Show(strBuilder.ToString())
        or MsgBox(strBuilder.ToString())
        firstTime = False
    End If
    Return strBuilder.ToString()
    Else : Return s
End If
End Function

```

**Example 2 :** Create a function and use it the function to display diff. background colors for the PARTYCODE.

1. CREATE A FUNCTION LIKE BELOW IN REPORT MENU->PROPERTIES->CODE

# Vinay Tech House

```
Public Function VinayTech_GetColor(ByVal DEPTNO As String) As String
```

```

IF DEPTNO = "10" then
    Return "PINK"
END IF
IF DEPTNO = "20" then
    Return "BLACK"
END IF
IF DEPTNO = "30" then
    Return "RED"
END IF
IF DEPTNO = "60" then
    Return "BLUE"
END IF

```

```
END Function
```

2. GOTO PARTYCODE textbox → rc text box properties and take fill color expression like below.,

```
= CODE::VinayTech_GetColor(Fields!PARTYCODE.Value)
```

## WORKING WITH .NET ASSEMBLIES (2 EXAMPLES)

1. Here that code which we create is used across multiple reports. (Global sharable).

2. We need to use .NET to implement this operation.

Process: 1. To reuse across reports the function (or) code should be deployed assembly  
2. We refer to the assemblies to work with the functions.

3. Functions are 2 types. a) static b) Non-static.

a) Non-Static Reference: a) Reference assembly b) Specify the instance name  
c) use the below exp. to call the inside report

= Code.InstanceName.method(argument)

b) Static Reference: a) Reference the assembly b) Directly call inside a report with the below syntax.

= Namespace.ClassName.method(argument)

Note: Static methods are good for performance. Since they are associated with class and not instantiated.

Practices: 1. Install visual studio (2008/09/10) in your system (.NET SW).

2. Open visual studio → file → new project → choose visual C-sharp template → class library → click ok.

3. Write the below functions inside the class.

```
public string MSBIMSG(string dayname)
{
    return "There is no class on:" + dayname;
}
public static string DEMOMSG(string dayname, string time)
{
    return "There is a new demo on:" + dayname + "@" + time;
}
```

Text Box 1: =Code.MyClass.MSBIMSG("MONDAY")

Text Box 2: ClassLibrary1.Class1.DEMOMSG("Wed", "8am")

171

## MSBI 2012 (SQL Server Integration Services)

Build → Solution →

Build → class library → Goto the class library solution folder → bin → debug → ~~copy~~ that DLL and paste in the below location of visual studio.

C → program files → visual studio 8/9/10 → common7 → IDE → private assemblies → paste it.

\* NOTE: 1. The lab practise copy the DLL from soft copies folder.

2. Create an instance like below.

Report menu → Report properties → References → browse to the folder where I copied → create instance name like below.,

class name  
ClassLibrary1.Class1

Instance name  
MyClass.

## EXPRESSIONS MORE EXAMPLES

### 1. Displaying every page 5 rows

To implement this in a group group on condition should be changed.

It is available in General section.

=CINT(CEILING(ROWNUMBER(NOTHING)/3)).

### 2. Displaying alternate color to alternate row

=IIF(ROWNUMBER(NOTHING) MOD 2, "Lime", "Tomato")

for every row there is a row number.

Row num	report mode -	Lime	Tomato
1	"	= 1	
2	"	= 0	
3	"		= 1
4	"	= 0	
5	"		= 1

### 3. Displaying every alternate three rows in different colors

=SWITCH (ROWNUMBER(NOTHING) MOD 3=0, "Lime",  
ROWNUMBER(NOTHING) MOD 3=1, "Tomato",  
ROWNUMBER(NOTHING) MOD 3=2, "Orange")

### 4. Repeating table header in every page

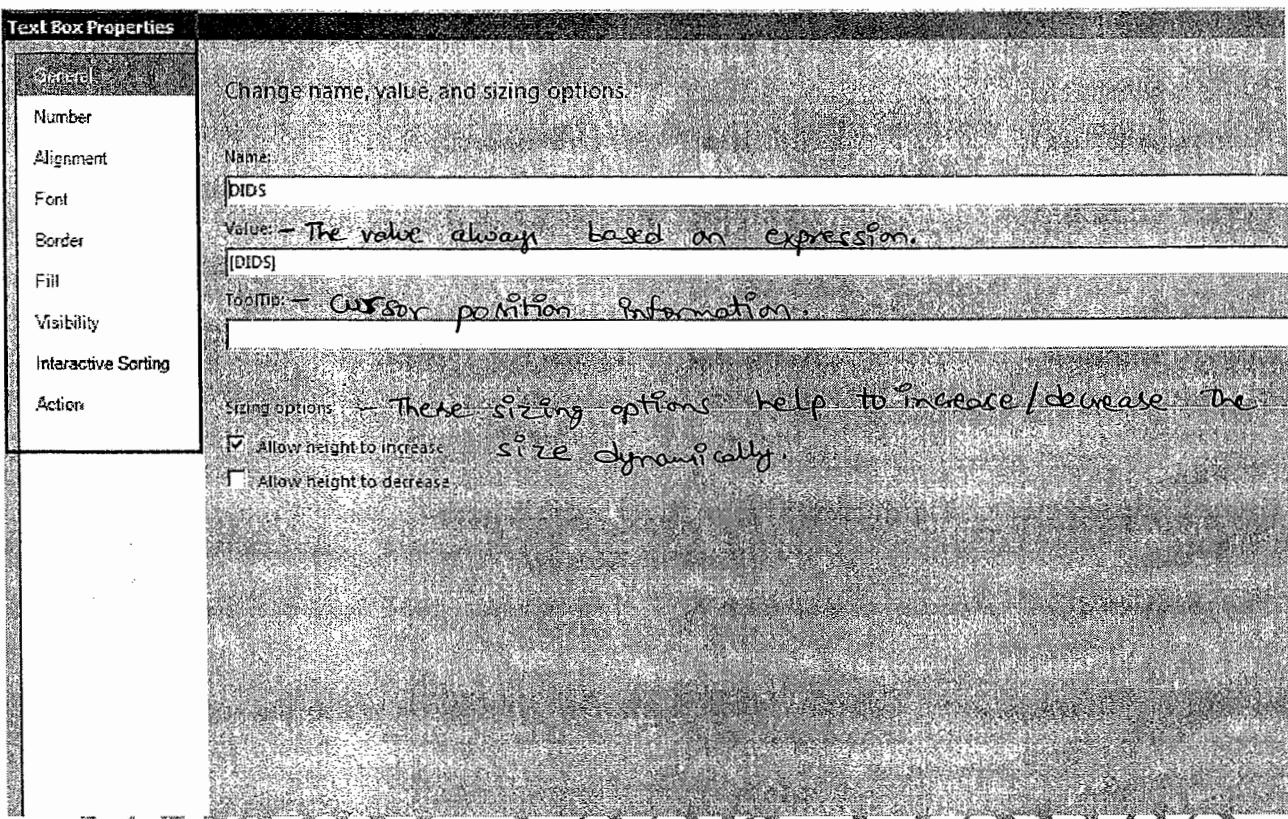
- Goto column groupings in the down and choose advanced mode.
- Goto row section choose first static option → F4 → fixed data=true , Repeat on new page=true .

\* 5. Display diff. sizes the income values. ( Less size below 30000 etc ).

Party Income textbox → rc → properties → font → click size expression and write below

## TEXTBOX PROPERTIES

A cell in a region (or) text box item <sup>in a</sup> toolbox is considered as Text box.



# Vinay Tech House

General : Name, value and Sizing Options

Number : Number and date Format Options available

Ex: Number → category section → currency → Decimal places → use 1000 separator  
 → Negative numbers → symbol → show symbol after value.

Alignment : Changing the text alignment and padding options (Horizontal/ vertical)

Font : Font, Size, Style, Color and Line spacing options available.

Creating hyperlink look : Italic → color (blue) → effects (underline).

Border : Border color, Style and width options available. To get border after settings we should click on outline.

Fill: Options related to Background colors and Images available

Visibility: Normal 4 visible options

**Interactive Sorting:** Support user interaction to sort the data dynamically at runtime.

Ex: Goto partitioning text box → Text box properties → Interactive sorting.  
 → enable interactive sorting → specify column name as, party sorting.

### **Actions:**

These additional operations we perform on the textbox values (leaving current boundary and going to other environment).

The below actions supported

- Goto URL
- Bookmark action
- Report action.

#### **1. URL action -** Calling a URL while analyzing the data.

Ex: Display the respective location information based on location click  
(or) Tap. Actions → Goto URL. → expression.

= SWITCH (Fields! PARTYLOC.Value = "HYD", "HTTP://www.HYDERABAD.COM",  
Fields! PARTYLOC.Value = "CHE", "HTTP://www.CHENNAI.COM",  
= "BLORE", "BLORE.COM")

Note: You will find hand icon only for specified item specified exp.

is called Bookmark Action.

#### **BOOKMARK ACTION:**

1. Take a table layout ,use eid,ename,eloc values in that.

Take eloc->alt+enter->BookMarkAction->loc

2. Take Chart report ,income in Data section and loc in category section.

Goto Series properties->action->Select Bookmark->loc ->ok

3. Now review the result and click the loc in the chart report then it opens the corresponding page in the table data region.|

#### **3. Report action -**

#### **Refer to the Drill Through Action Concept**

**VARIABLES**

Holds a value, the value can be changeable. (Normal dt)

In SSRS, the value we can use for our intermediate operations.

There are 2 kinds of variables. (a. Report variable b. Group variable).

- A) Report Variable: 1. created inside report, used across all items of report.  
2. we can't use these in calculated fields.

Creation: Report Menu → Properties → Variables → add variable →

Name: PINC Expression: = 12/100

Usage: Goto Party Income textbox → rc → expression → = fields!PartyIncome \* PINC

- B) Group variable: 1. created inside group and the scope also at group level.

Creation: Goto Group properties → variables → add

Name: pinc1 expression: = 12/100

Usage: Goto party income textbox → rc → expression →

**Vinay Tech House**

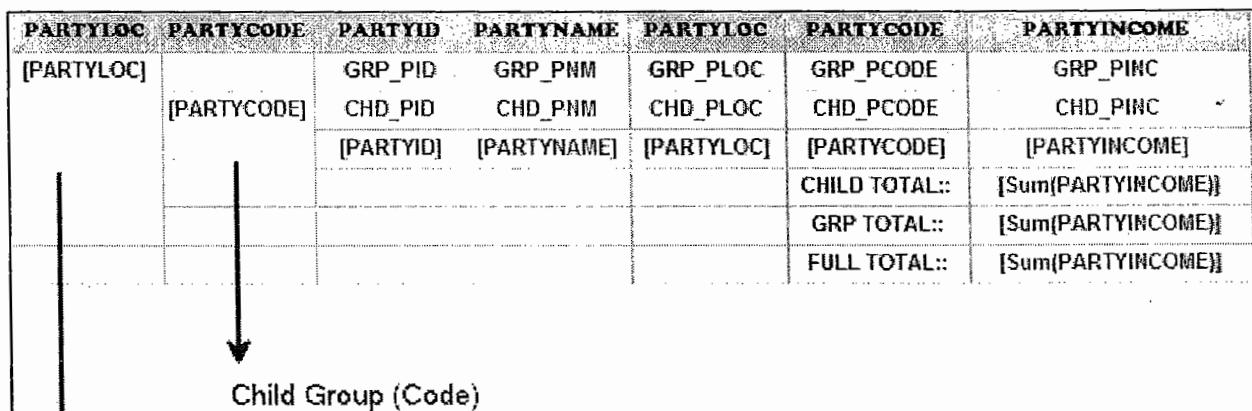
Note: Report variables you can find in the expression section where as group variables we can't find, so we have to type manually.

summary or go for groups to improve the data processing in case of condition.

## WORKING WITH GROUPS

1. To classify the database on Condition.
2. To display the content in multiple pages.
3. To show aggregations more easily, groups are helpful.

### Real Time Usage :



The screenshot shows a report viewer with the following details:

Report Name: Reports  
Execution Time: 5/15/2012 10:17:37 AM  
User ID: RAVAN\Administrator

Table Data:

Page Number : :2-TO-3			Page Sum: 120000				
HYD	10	GRP_PID	GRP_PNM	GRP_PLOC	GRP_PCODE	GRP_PINC	
		CHD_PID	CHD_PNM	CHD_PLOC	CHD_PCODE	CHD_PINC	
		1	VINAYAKA	HYD	10	70000	
	20	CHD_PID	CHD_PNM	CHD_PLOC	CHD_PCODE	CHD_PINC	
		2	VINAY	HYD	20	50000	
					CHILD TOTAL::	70000	
					CHILD TOTAL::	50000	
					GRP TOTAL::	120000	

### TAKING PARENT GROUP

Details Section → Addgroup → Parent group →

Name: Party Loc

- Add Group Header
- Add Group Footer

**TAKING GROUP TOTAL**

Goto group footer → Party Income column → Select party Income

You will get sum of income by default.

**TAKING CHILD GROUP**

Goto Details section → Add Group → Child Group → Name: PartyCode

Add Group footer

**TAKING CHILD TOTAL**

Goto child Group footer → Select party Income in party Income column

You will get sum of income

**TAKING FULL TOTAL**

Goto parent group footer section → Select party Income in party Inc

column. You will get sum of income.

+c → Insert row → outside group → below.

**GROUP PROPERTIES**

a) **General** – Here name and group expression will display \*This expression contains a normal col. name (or) a conditional expre

b) **Page Breaks** –

- Between each instance of a group.
- Also at the start of a group -- Before first group an empty page
- Also at the end of a group -- After last group an empty page

c) **Sorting** – Groups display in the specified sort order.

Ex: i) Locations in descending order, within the locations <sup>names</sup> in ascending order. (Multi-column Sorting).

i) Group properties → sorting → location descending order.

ii) Table properties → sorting → name in ascending order.

d) **Visibility**

e) **Filters** – To take only required groups this option helpful.

f) **Variables** – Group variables, use only within the group.

Ex:

g) **Advanced** – i. These features help to locate a particular group easily and to implement hierarchical operations.

**Document Map**: - The assigned column system shows in the left hand side panel for quick reference.

## IMPLEMENTING RECURSIVE PARENT

- It helps to implement hierarchical analysis.

In a table we have multiple columns, and there is an indirect relationship b/w the columns recursive parent is helpful.

Eg: from the sales table display the relationship b/w emp and mgr in the hierarchy partition.

Recursive Parent: 1. Take a table report item.

2. Use ename, manager name, sal in the table.

3. Row group → Add

General group → Ename

Recursive parent → : Mgrname

Visibility:

Hide

Toggle item → Ename

$$\text{Cstr}(2 + \text{level}(\text{l}) * 10) + "pt"$$

$$\text{Level 0} = 2 + 0 * 10 = 2$$

$$\text{Level 1} = 2 + 1 * 10 = 12$$

$$\text{Level 2} = 2 + 2 * 10 = 22$$

- Goto Ename textbox → TextBox Properties → Alignment → Padding left → write the below expression.

$$= \text{Cstr}(2 + \text{level}(\text{l}) * 10) + "pt"$$

## PARAMETERS, THEIR TYPES (4) AND EXAMPLES

Parameters are required for 2 operations. a) To parse values which controls the report b) for better user interaction.

There are 5 types.

- non-queried parameter
- queried parameter
- multivalued parameter
- cascade parameter
- entering values manually at run-time.

Blind steps to work with parameter:

- Create a parameter
- Use the parameter in the dataset. (for single value with equals operator and multiple values IN operator).

## PARAMETERS and VARIABLES

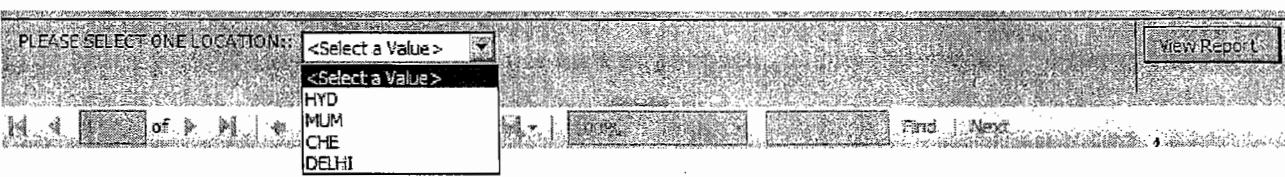
### VARIABLES

- \* Holds a value
- \* Scope in report/group
- \* Can be changed
- \* No user interaction
- \* We will not use in dataset level to filter
- \* Unknown values

### Look and Feel Of Simple Parameter Passing:

### PARAMETERS

- \* Display set of values
- \* At report level
- \* No change
- \* Support user interaction
- \* We can't use at dataset level.
- \* Known values



### 1. Non-queried

### PARAMETERS

### 1. For static elements

operation this process is recommended.

2. Here we enter user values manually. To show in the dropdown list.

VIEW->REPORTDATA->

1. PARAMETERS->RC->ADD PARAMETER  
GENERAL NAME :VARLOC

PROMPT :PLEASE SELECT THE LOCATION:

DATA TYPE :TEXT

SET PARAMETER VISIBILITY

SELECT VISIBLE

### AVAILABLE VALUES

### SPECIFYVALUES

#### ADD THE BELOW

LABEL VALUE

HYDERABAD HYD

CHENNAI CHE

BANGALORE BLORE

DEFAULT VALUES -> NO DEFAULT VALUE

### 2. DATASET->RC->DATASET PROPERTIES

QUERY-> ADD THE WHERE CONDITION

WHERE P.PARTYLOC = @PARLOC

OK → BROWSE AND SEE THE RESULT

**2. Queried****PARAMETERS**

1. In case the locations change

dynamically this option is useful.

2. For this we should create a dataset to pull the data from a query.

1. CREATE A NEW DATASET TO TAKE ONLY DISTINCT LOCATIONS VALUES

VIEW-&gt;REPORTDATA-&gt;ADD DATASET-&gt;

NAME : DSTLOC

QUERY : SELECT DISTINCT(PARTYLOC) FROM LOCATION

## 2. CREATE A PARAMETER BASED ON THE ABOVE DATASET RESULT

PARAMETERS-&gt;RC-&gt;ADD PARAMETER

GENERAL NAME : VARLOC

PROMPT : PLEASE SELECT THE LOCATION::

DATATYPE : TEXT

SET PARAMETER VISIBILITY

SELECT VISIBLE

## AVAILABLE VALUES

GET VALUES FROM QUERY

DATASET : DSTLOC

VALUE FIELD : PARTYLOC

LABEL FIELD : PARTYLOC

DEFAULT VALUES -&gt; NO DEFAULT VALUE

## 3. USE THE ABOVE PARAMTER IN THE ACTUAL REPORT DATASET (WHICH IS DISPLAYING VALUES IN THE REPORT)

DATASET-&gt;RC-&gt;DATASET PROPERTIES

QUERY-&gt; ADD THE WHERE CONDITION

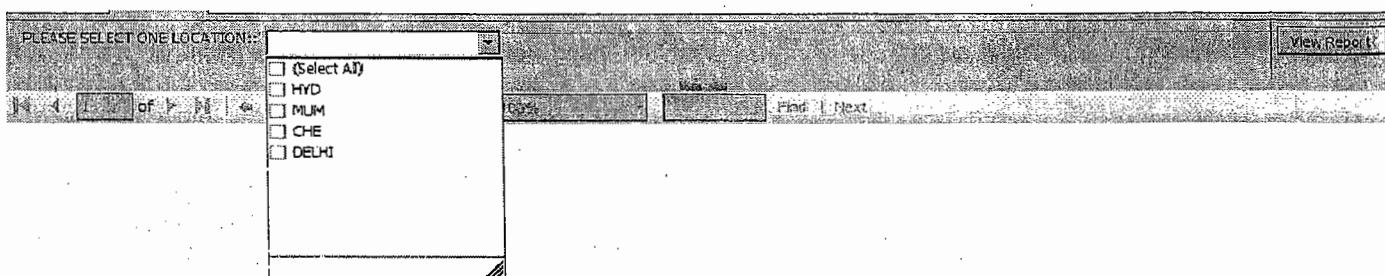
WHERE P.PARTYLOC = @PARLOC

OK

**3. MULTIVALUED****PARAMETERS**

1. Here users will get a chance

to choose multiple values (all values).



Practices: To implement this 2 options required. a) In the parameter general section choosing allow multiple values. b) Where clause filtering with IN operator.

#### 4. Cascade.

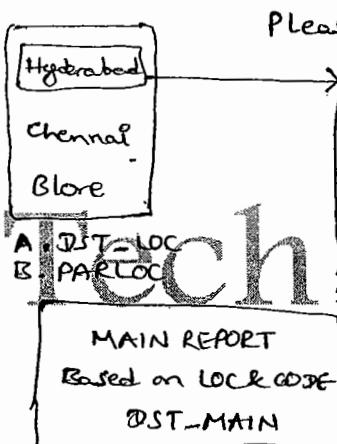
#### PARAMETERS

Based on one parameter it display other parameter information then it is cascaded.

Ex: Displaying locations first, after choosing a location displaying codes, after choosing code displaying location and code information.

#### QUERIED CASCADING:

Please select the location



Please select code

# Vinay Tech House

#### 1. CREATE A NEW DATASET TO TAKE ONLY DISTINCT LOCATIONS VALUES

VIEW->REPORTDATA->ADD DATASET->

NAME :DSTLOC

QUERY :SELECT DISTINCT(PARTYLOC) FROM LOCATION

#### 2. CREATE A PARAMETER BASED ON THE ABOVE DATASET RESULT

PARAMETERS->RC->ADD PARAMETER

GENERAL NAME :VARLOC

PROMPT :PLEASE SELECT THE LOCATION::

DATAATYPE :TEXT

SET PARAMETER VISIBILITY

SELECT VISIBLE

#### AVAILABLE VALUES

GET VALUES FROM QUERY

DATASET :DSTLOC

VALUE FIELD :PARTYLOC

LABEL FIELD :PARTYLOC  
 DEFAULT VALUES -> NO DEFAULT VALUE

### 3. CREATE A NEW DATASET TO TAKE ONLY DISTINCT CODES BASED ON THE LOCATION

VIEW->REPORTDATA->ADD DATASET->  
 NAME:DSTCODE  
 QUERY:SELECT DISTINCT(PARTYCODE) FROM PARTY

### 4. CREATE A PARAMETER BASED ON THE ABOVE DATASET RESULT

PARAMETERS->RC->ADD PARAMETER  
 GENERAL NAME :VARCODE  
 PROMPT :PLEASE SELECT THE PARTYCODE:  
 DATAATYPE :TEXT  
 SET PARAMETER VISIBILITY  
 SELECT VISIBLE

#### AVAILABLE VALUES

GET VALUES FROM QUERY

DATASET :DSTCODE

VALUE FIELD :PARTYCODE

LABEL FIELD :PARTYCODE

DEFAULT VALUES -> NO DEFAULT VALUE

# Vinay Tech House

### 5. USE THE ABOVE PARAMTER IN THE ACTUAL REPORT DATASET(WHICH IS DISPLAYING VALUES IN THE REPORT)

DATASET->RC->DATASET PROPERTIES

QUERY-> ADD THE WHERE CONDITION

OK

### 5. PARAMETERS ADDITIONAL

Entering values manually.

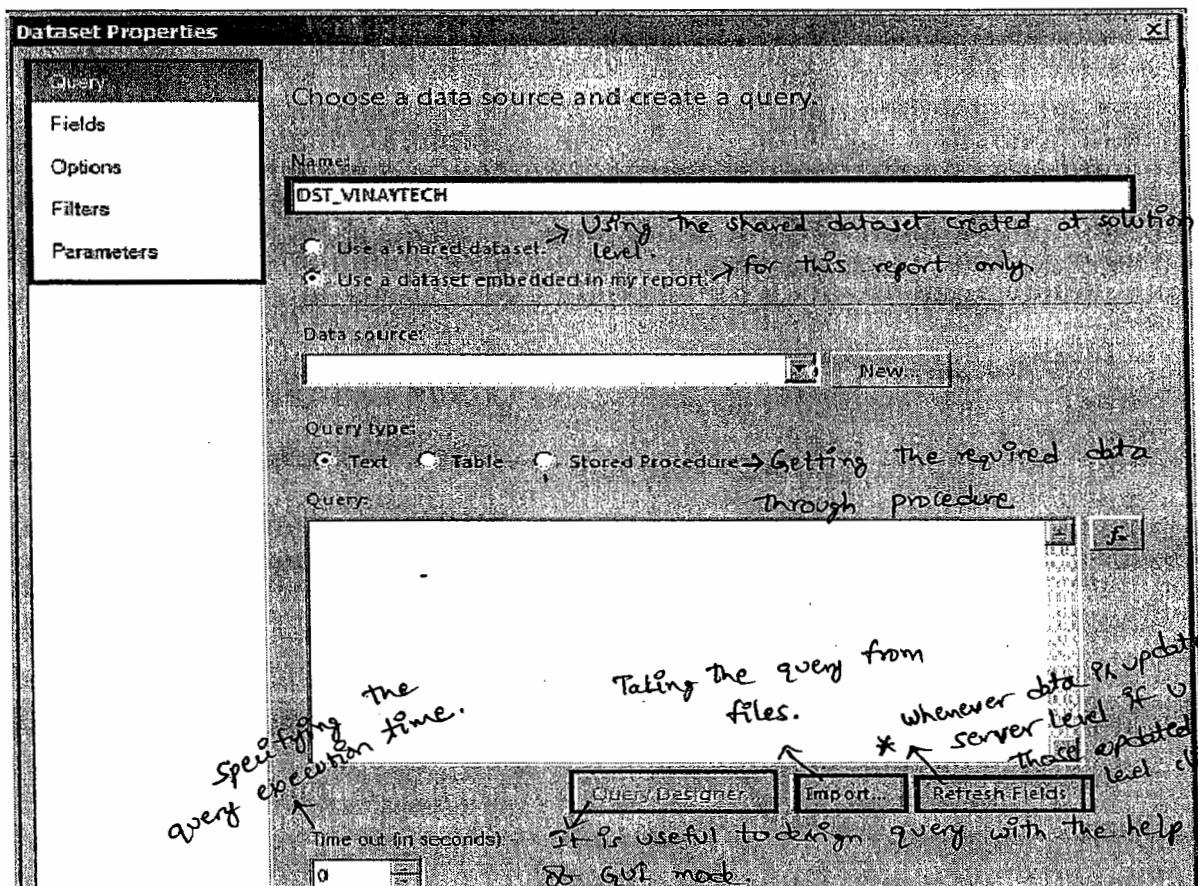
In this situation a blank display to enter a value manually.

Practice: Goto parameter → Available values → specify none.

- FAQ's:
1. How do you allow user to choose multiple values.
    - a) By using Multivalue parameter.
  2. How do you set a default element always.
    - a) Goto default value and specify the value.

## DATASET PROPERTIES

- Generally dataset takes a table (or) a query (or) stored procedure to retrieve the information.



Options: Talk about collation, case sensitivity, accent sensitivity, subtotal etc. operation

Case sensitivity false      Accent sensitivity

RAMA = rama

Hour = our

\* Filters: \* The data collected in the dataset.

It supports different perceptions on the same data.

System will not apply filter directly on source.

Parameters: 1. To pass values to stored procedure value.

2. To operate query with parameters this is helpful.

- it would give good performance by eliminating processing and writing the calculation multiple times.

4. **Support parameters**

**MSBI 2012 (SQL Server Integration Services)**

two times utilization for the specified calculation.

No support to parameters. 183

# **QUERY & CALCULATED FIELDS**

Query field : 1. It helps to create a copy from the existing columns.  
2. NO calculations supported.  
3. The greatest advantage is it keeps the original one and helps to work on the copy of the column.

Calculated field: 1. Support calculations.

2. It takes source database formulas for its operation.
  3. This calculations applied always source level so report data retrieve time might increase.

Eg: Creating party income1 query field , party income like calculated field with an increment of 10%.

Navigations fields → add → query field → Field Name Field Source  
PARTY INCOME1 PARTY INCOME → click add.

→ Calculated field → Field Name Field Source  
PARTY INCOME-MIKE PARTY INCOME \*20/100  
CALCULATED FIELD and REPORT CALCULATION.

Q: What is the diff. b/w Calculated field Report Textbox Calculated field  
A: 1. No variables supported Supported  
2. Apply operations during retrieval During report process (after retrieval)

## **DRILL DOWN REPORTING**

1. Within the same report if we provide navigations with expansion collapse and showing/hiding toggling is helpful.
  2. Generally for top-down and bottom-up analysis this deep drill down operation is helpful.
  3. To implement this we need toggling.

Eg: Display Gender when expanding location, display table data when expanding Gender

# **DRILL THROUGH REPORTING**

1. Moving from one report to another report based on some action is called Drill through reporting.
  2. This action can also called Report action.

Eg: Displaying a customer detailed information in report called party-report when there is a click on partyID in the report1.

## PARTY - REPORT

1. Create parameter VARID by giving a name (No additional options we need to set)
  2. Use it in the data set where

PARTY 1 - REPORT 1

1. Take a table and set PARTYID, PARTYL and INCOME values.
  2. rc → partyid textbox → Textbox prop.  
→ actions → goto report → specify a report PARTY-REPORT → Name Value  
VARID PARTYID

# Vinay Tech House

## **CONNECTING TO CUBE TO CREATE REPORT**

- \* Datasource → Add datasource → ② Embedded connection  
Type  
MS SQL Server Analysis services

click on edit → provide SSAS server → connect to a database  
select or enter a db → OK →

dataset → Add dataset → click on query designer → drag and drop Color sales and Tax amount → OK.

Product	Category	Sub-Category	Region	Period	Segment	Promotions	
Food			Multi Select				
Gender	Education Level		Store Sales	Store Cost	Profit	KPI	Face
F	Bachelors Degree		927.210	357.28	559.93		
	Graduate Degree		132.780	54.16	78.62		
	High School Degree		1116.660	436.70	682.96		
	Partial College		513.440	206.18	307.29		
	Partial High School		1379.330	563.77	815.56		
M	Bachelors Degree		1128.220	432.59	679.59		
	Graduate Degree		411.210	169.71	241.50		
	High School Degree		1263.780	507.45	761.33		
	Partial College		333.950	126.21	207.74		
	Partial High School		1184.590	471.77	712.82		

### CREATE REPORT FROM PROCEDURE

Create Proc PR1

as

begin

select \* from emp

end

1. Create data source with the help of emp database.
2. Dataset → Add dataset → Query type → click on OK  
③ Stored procedure

**CREATE REPORT FROM AN EXCEL SHEET****CHART REPORT**

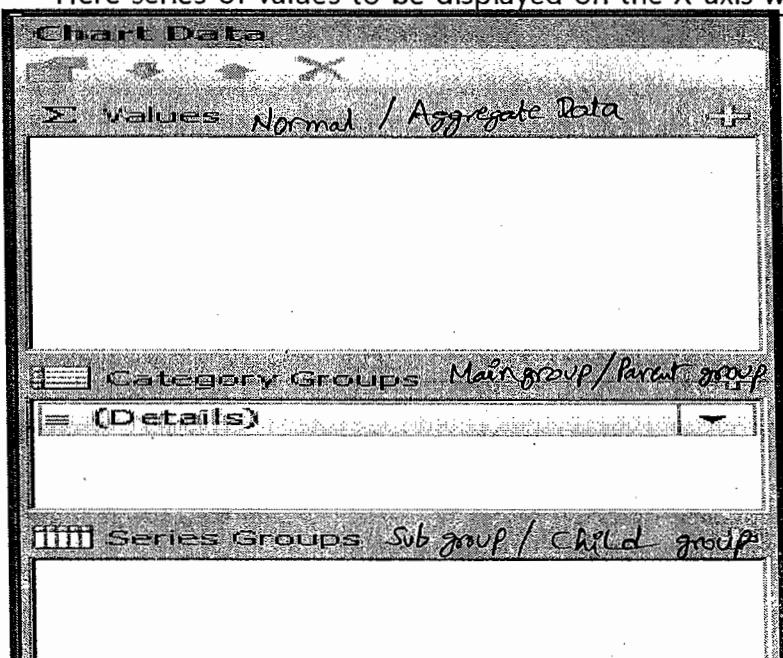
It contains Three major sections.

**A) Data Section:** -

Here the actual data or summarized data (aggregated data) presented here.

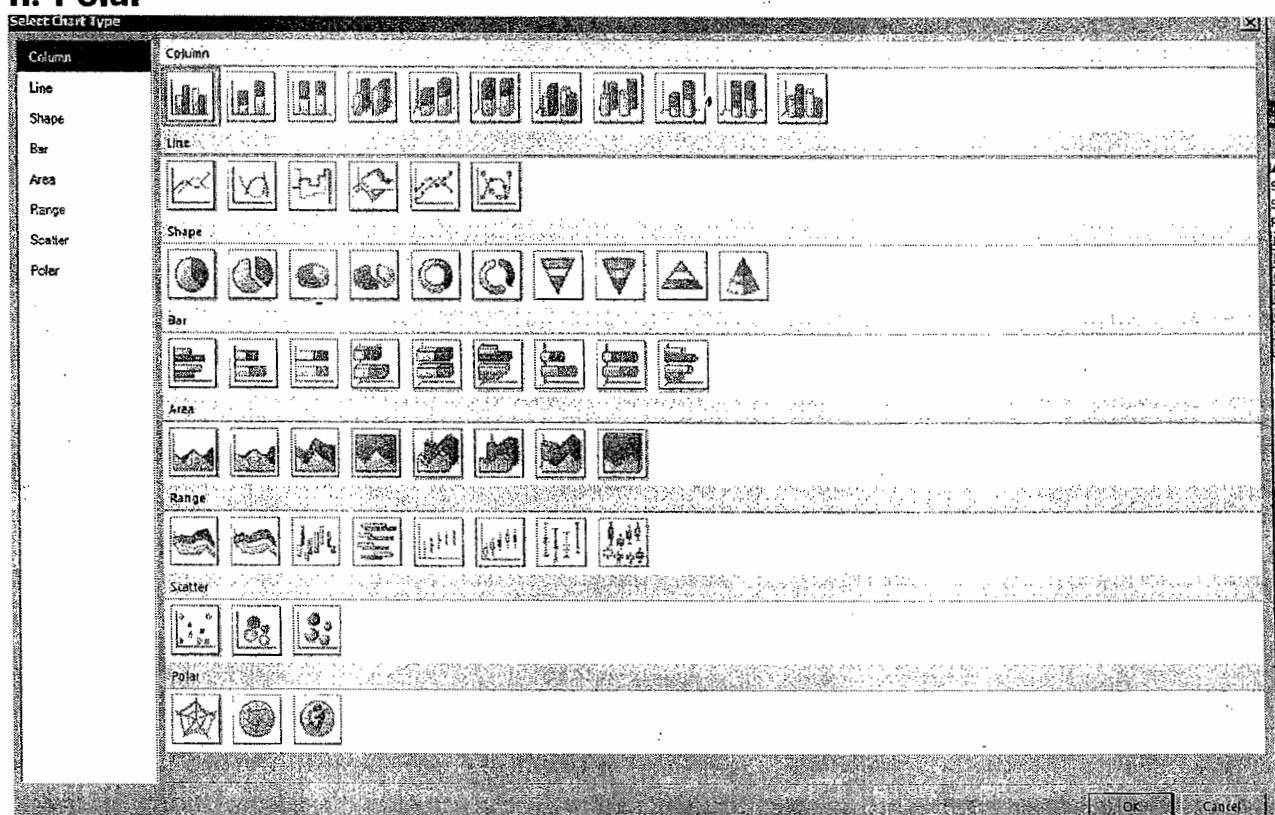
**B) Category Section: -** Here category or group by fields data displayed.**C) Serialized Section: -**

Here series of values to be displayed on the X-axis will be taken.



## Chart Types:

- Column
- Line
- Shape
- Bar
- Area
- Range
- Scatter
- Polar

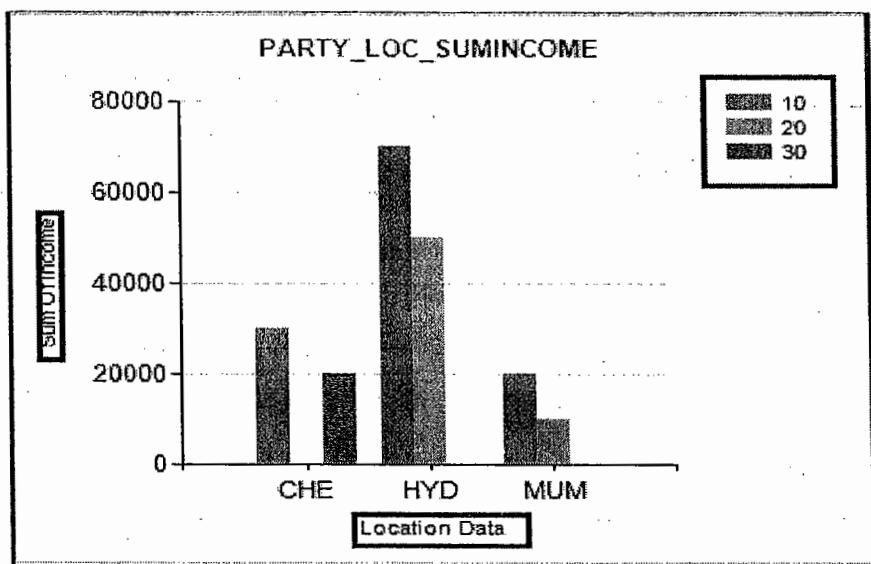


### Example 1: Display Locations and their <sup>Job</sup> <sub>Salary</sub> partyincome total

Double click on chart it will open chart data at  $\Sigma$  values click a "+" icon take salary. Category group click on "+" icon take job → click on preview

### Example 2: Display locations ,in each location depts and their <sup>Job</sup> <sub>Salary</sub> total income

- $\Sigma$  values - Sal
- Category group - Job
- Series group - Deptno

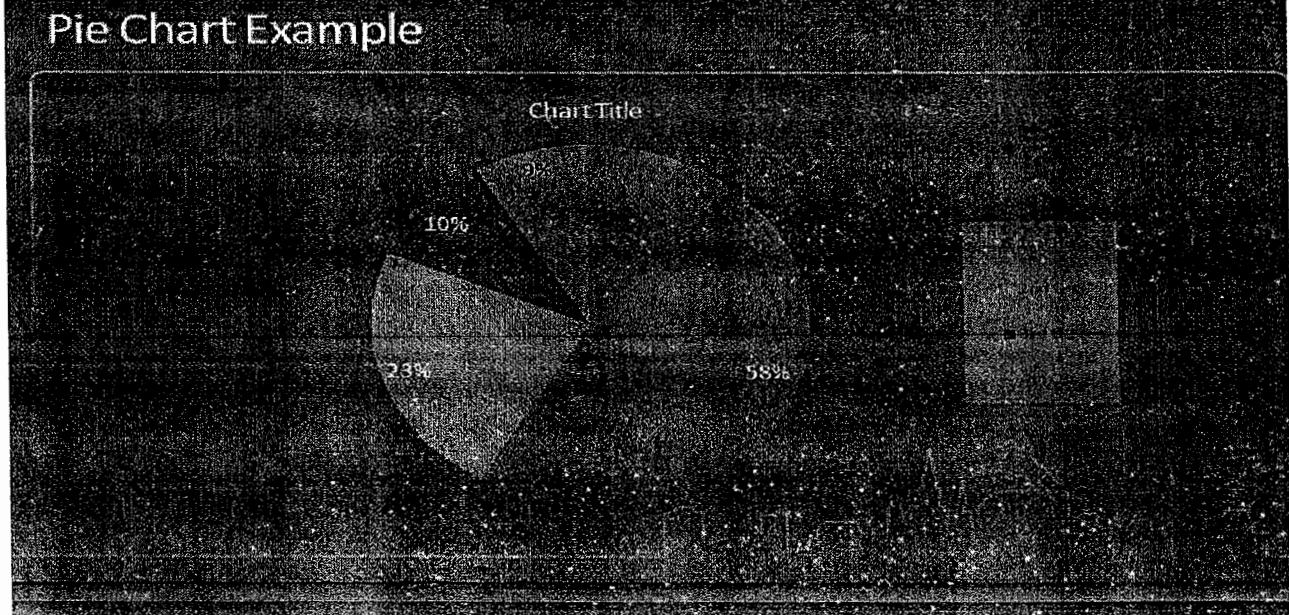


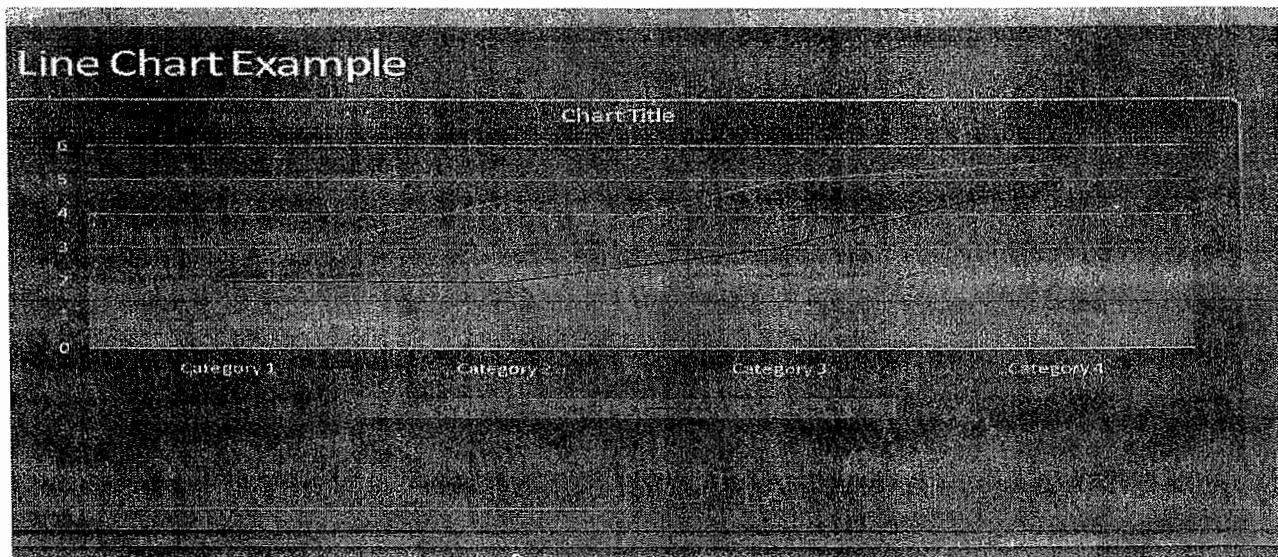
Note, Right click on the chart box show data labels.

### Changing Chart Type (Shapes, Exploded Shape) & Features:

- R-click on chart → change chart type
- By default it displays aggregate information.
- To change agg info into percentage
- Goto labels → series label properties → label data (# per cent)

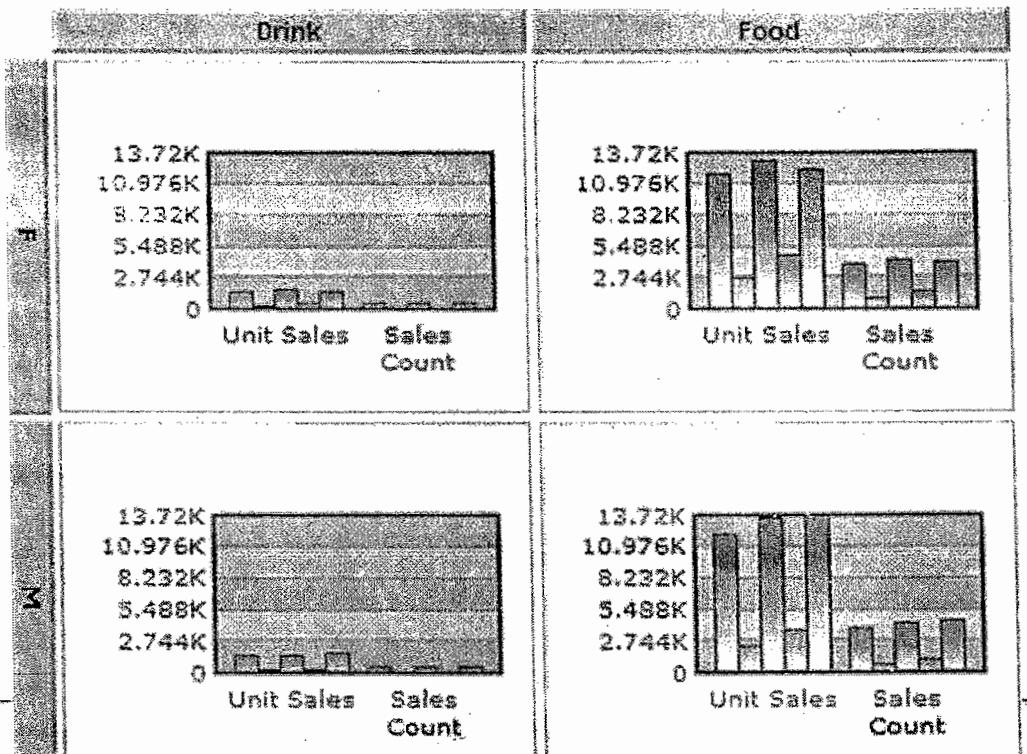
Pie Chart Example



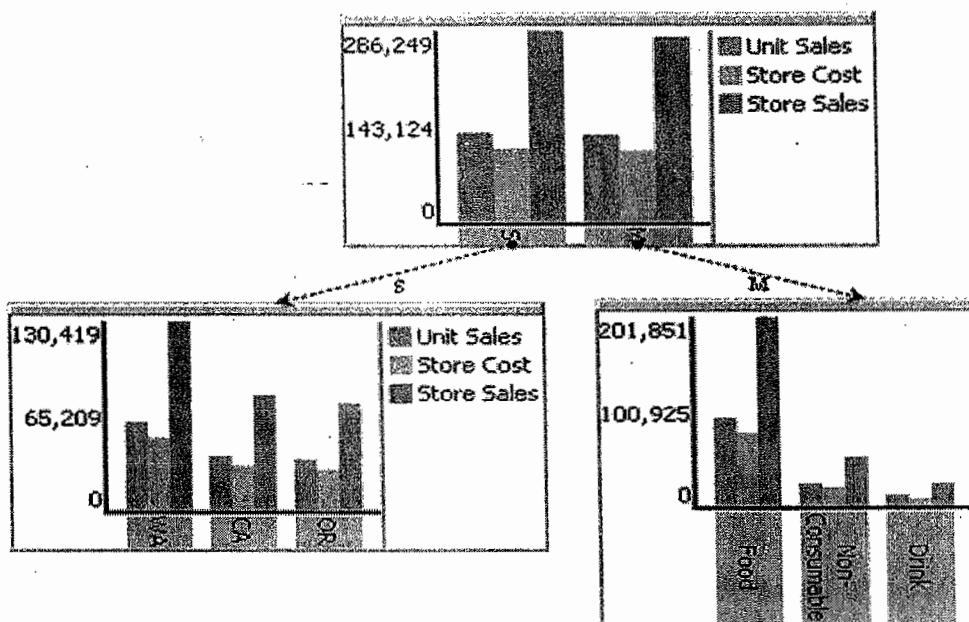


## Exercises

# Chart Grid Report



## Bar-Chart Tree Report



### CHART AREAS PROPERTIES:

- RC on Chart area → Chart area prop.
- 3D options
- visibility
- fill
- Border
- Shadow

### CHART PROPERTIES

Chart properties → Series properties → Series data → visibility, legend

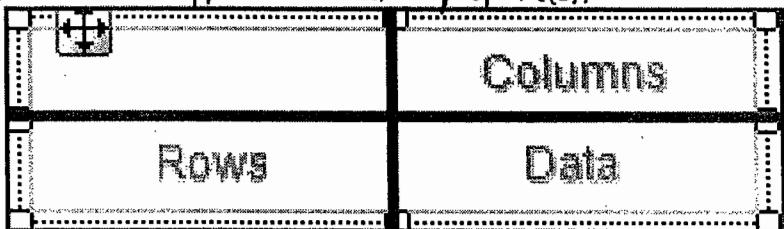
### LEGEND PROPERTIES:

Select legend rc → legend properties → general, visibility, font, fill, border, shadow.

**MATRIX REPORT**

→ Cross tab operation

- Displays normal or aggregated data.
- It has three sections.
  - Row selection → Takes row wise information.
  - Column selection → Display column wise info.
  - Data selection → Aggregated or normal data
  - Matrix support tablix properties.



Ex: Insert → Matrix → column (partyloc) → In row - Add group → row group → parent group → group by (partycode),  Add group footer.

OK.

data → party income

**Example 1: Display Statewise ,Sales total income**

Row wise → select the state, Column wise → select dept data wise →  
select the sales.

**Example 2; Statewise districtwise deptwise and sub deptwise total income**

RC matrix → row wise districts → data wise → sales income column wise → dept code

		SALES		HR	
		10	20	10	20
AP	VSKP	10000	20000	20000	10000
	RR	20000	30000	40000	10000
UP	ALLAH	30000	20000	30000	20000
	LAD	20000	30000	20000	20000

Select entire column dept code → rc → Add group → parent group → dept → ok

→ goto row groups → districts → Add group → parent group → state → ok

**Example3 : Display total income row wise and column wise**

Select dept code column → right click → insert column → outside group right  
 select entire row → insert row → outside row below

	SALES	HR	SINC
AP	20000	20000	40000
UP	10000	20000	30000
FULL TOTAL	30000	40000	70000

4. It supports table properties.
- # Vinay Tech House
- LIST REPORT**
- 1. Mainly designed for free from reporting.
  - 2. Here we can design have our own layout. we don't have any fixed layout.
  - 3. Generally train tickets, movie tickets, voter list, PAN card created by list report.

	# 505, 5th Floor, Annapurna Block Ph: 9573168449, 66638869
Student ID :	V_001
Course :	MSBI 2012
Fee Paid :	8000
Due DT:	2 DAYS(12TH)
Due Amt:	500

**RECTANGLE PROPERTIES:**

## 1. General

Visibility

Fill

Border

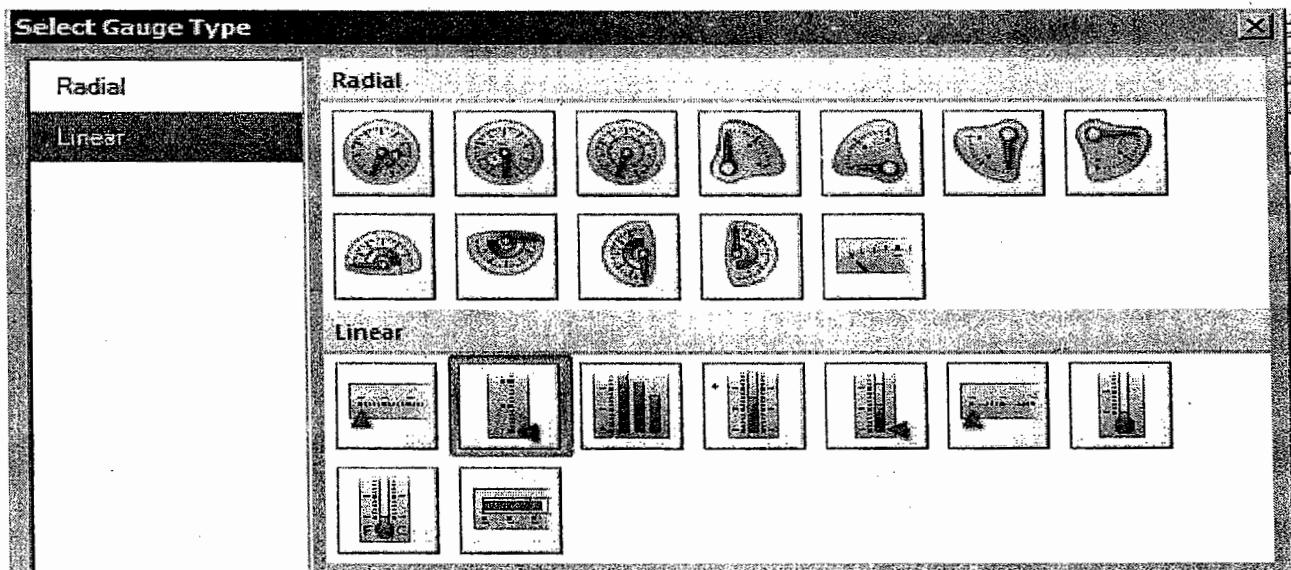
Page break options

- Add a page break before - It will add empty page before report
- Add a page break after
- Omit border on page break
- keep content together on a single page, if possible

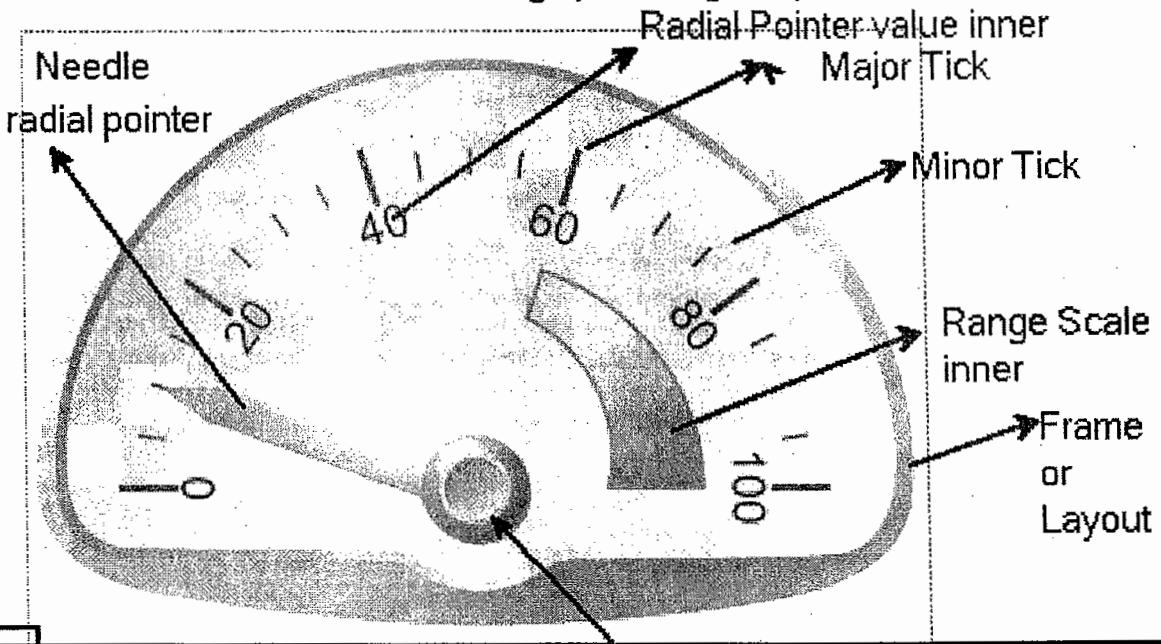
# Vinay Tech House

**GAUGE REPORT**

Displays only one value at a time.  
Means single value. ( $\frac{10}{10}$ )



View

Radial Gauge (180 degrees)

**Note:** To work with any item go to the below user friendly properties

a) Pointer Properties

b) Scale Properties c) Gauge Properties

Pointer Cap

Pointer properties: select the pointer → rc → Pointer properties.

i) pointer fill → fill style → color  
     Solid gold

ii) Gap fill → color  
    Green

Scale properties → select the scale → rc → scale properties

i) Label font → style → size → color  
     Bold 24 pt yellow

ii) Major tick marks → width percent

iii) Minor tick marks → length percent

iv)

\* Gauge properties → select gauge

i) Back fill

ii) frame fill

iii) frame - within the frame we have glass

## INDICATORS

table (or) matrix.

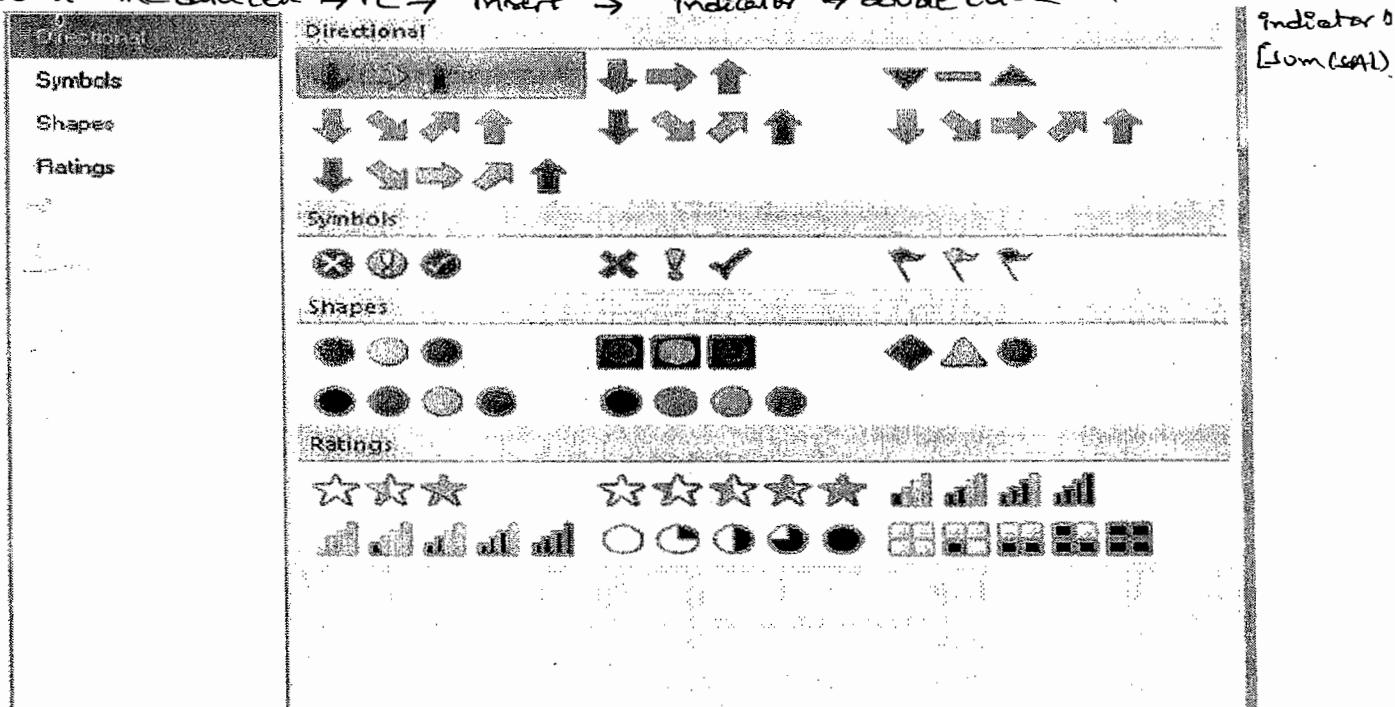
→ Indicators designed for range of values.

1. These are simple charts to convey lot of info. in single place.
2. These are minimal gauges that convey the state of single data value.
3. These are simple and usually effective when we present numerical info. in a graphical format.

Navigation: Create emp report with the help of table.

Select last column → Insert column right

Select the data cell → re → insert → indicator → double click on indicator.



Select indicator → re → indicator prop. → click on values and states.

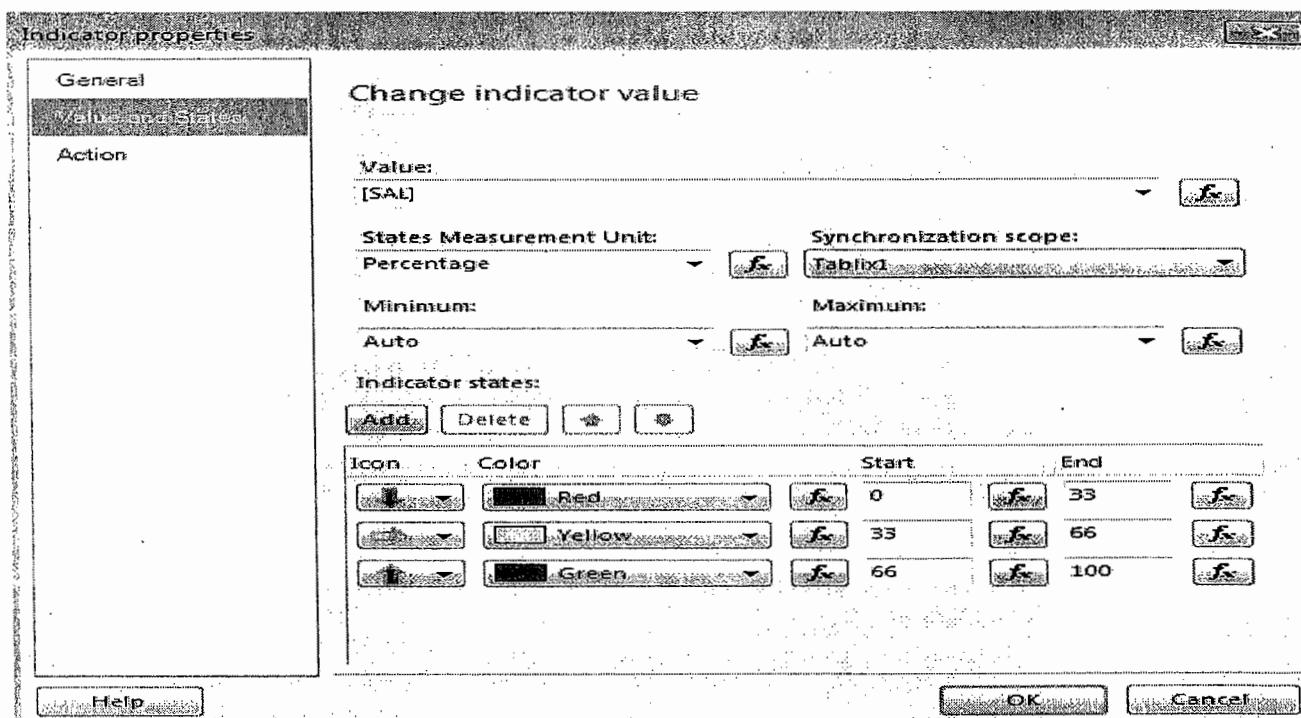
states measurement unit

Numeric

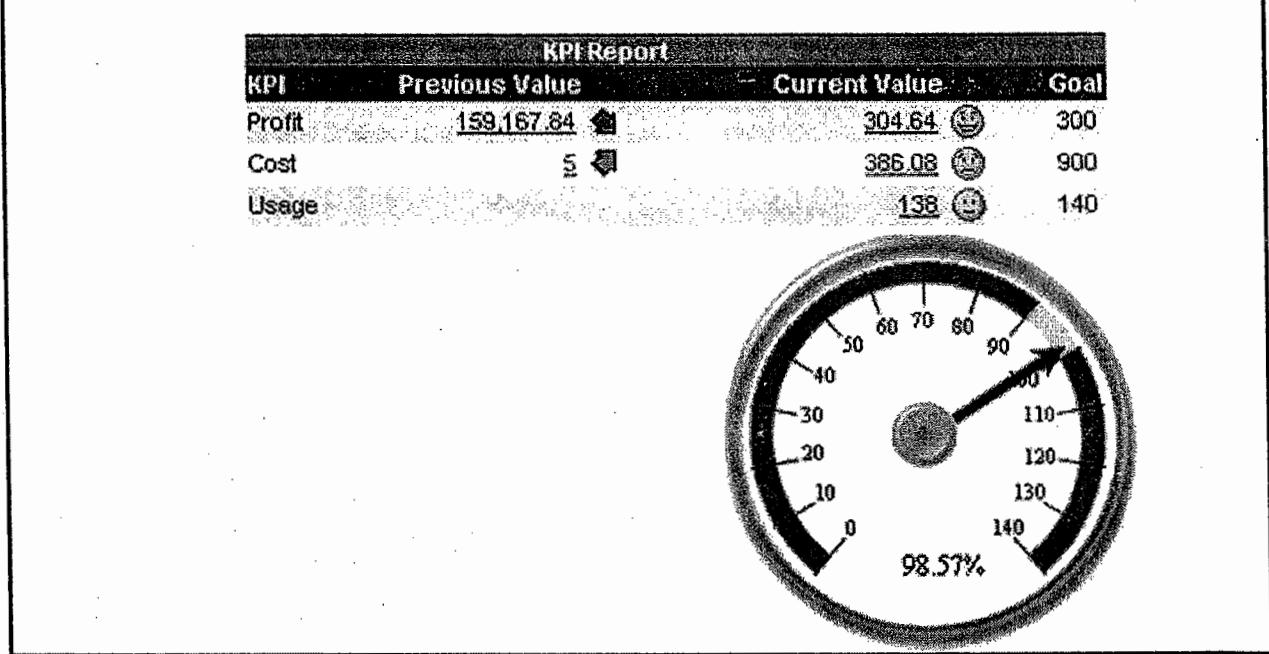
Percentage

Numeric

1. We can able to provide our own range values (start and end values)
2. We can provide more number of ranges.



## Key Performance Indicator Report



## Analysis Services KPI Report

Adventure Works							
Visible	Report	Name	Value	Goal	Status	Trend	Info
<b>Financial Perspective</b>							
<b>Maintain Overall Margins</b>							
<input checked="" type="checkbox"/>	Adventure Works	Net Income	\$12,609,503.00	\$5,583,900.00			
<input checked="" type="checkbox"/>	Adventure Works	Operating Profit	\$16,728,234.50	\$5,583,900.00			
<input checked="" type="checkbox"/>	Adventure Works	Operating Expenses	\$27,681,868.50	\$6,264,750.00			
<input checked="" type="checkbox"/>	Adventure Works	Financial Gross Margin	\$44,390,103.00	\$11,848,650.00			
<b>Grow Revenue</b>							
<input checked="" type="checkbox"/>	Adventure Works	Return on Assets	0.917673375801342	Not Budgeted			
<input checked="" type="checkbox"/>	Adventure Works	Financial Variance	0	→			

### DATABARS AND SPARKLINES

→ Multiple values in single location.  
(10, 20, 30, 40, 66 --)

\* Sparklines and databars are small, simple charts that convey a lot of information in a little space.

\* Sparklines and databars are often used in tables and matrices. The impact comes from viewing many of them together, and being able to quickly compare them one above the other, rather than viewing them singly.

\* They make it easy to see the outliers, the rows that are not performing like the others. Although they are small, each sparkline often represents multiple data points, often over time.

Sparkline navigation: 1. Create data source with the help of <sup>any</sup> database  
2. Create dataset with the help of hardcore values.

Eg: select 101 as ID , 'BALAJI' as NAME

Op: 101 BALAJI

Union

102 VAMS

Select 102 as ID , 'VAMS' as NAME

103 KUMAR

Union

select 103 as ID , 'KUMAR' as NAME

3. Take the matrix report → row wise product info → column wise sales date → data wise sales.

Vinay TechHouse

# 505, Annapurna Block, Aditya Enclave, Ameerpet, Hyd. Ph: 04066638869/9573168449

4. Select entire ~~group~~ column → rc Insert column → outside group select

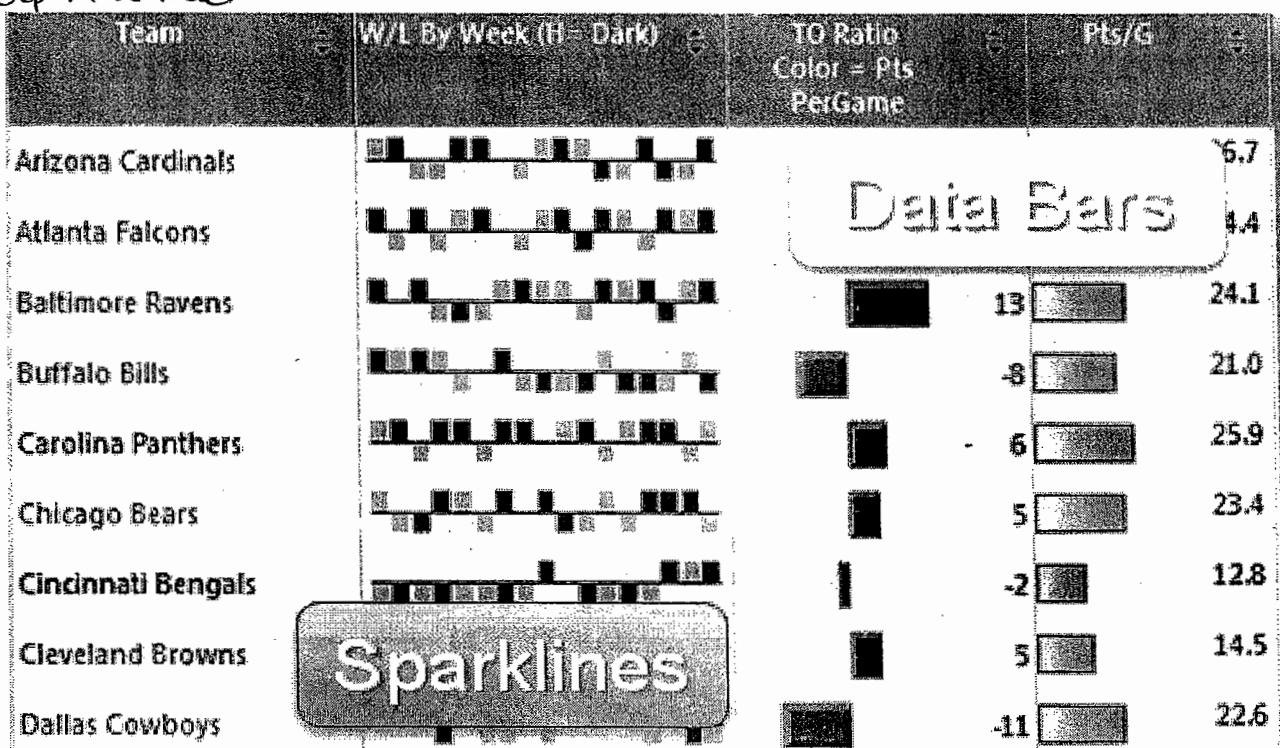
5. In dataset → rc → Insert → sparkline → select the data to late

6. Double click on sparkline

7. Enable show data labels.



- \* Databases can represent multiple data points, but typically illustrate only one.
- \* Sparklines and databases have the same basic chart elements & categories, series and values, but they have no legend, axis lines, labels or tick marks.
- \* Each sparkline typically presents a single series. You can't add a sparkline to a detail group in a table.



Number of Visits By State Data Bar

State	Visit Count
California	597
New Mexico	387
Texas	342
Connecticut	322
Michigan	322
South Carolina	289
Arizona	277
Idaho	265
New Hampshire	261
Alabama	259

\* Because sparklines display aggregated data, they must go in a cell associated group.

\* Select product dataset → rc → insert outside group below.

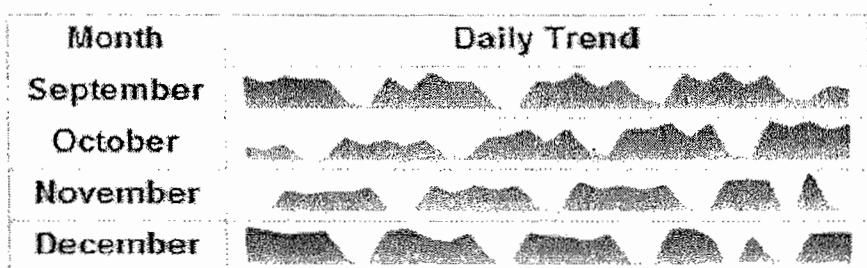
\* select entire column → insert column outside group right

\* Take header section → re → insert → database

double click on data bars.

category groups → enable show data labels.

## Visits Trend Analysis (Sparkline)



### MAP REPORT

1. Creates MAP from Geo spatial coordinates.
2. There are 2 important <sup>items</sup> reports in map reporting.
  - a) MAP report item (New report item for map reports)
  - b) MAP layer (Displays the Geo graphical background (or) map element based on Geo spatial data).

When we add map to the report first layer element added base on spatial data. We can change the layer later by going to new map layer → Add map layer options.

The Below types of layers you can add to the report.

- a) Polygon → represents graphic areas such as countries, states, cities etc.
- b) Line → represents paths (or) routes
- c) Point → represents locations such as stores, cities (or) places.
- d) Tile → represents microsoft bing map tiles in map report background spatial data we get from three sources.
  - a) From map gallery - which is having US data in the SSRS default gallery
  - b) ESRI Shape files (Environmental System Research Institute)
  - c) Spatial data.

*It is also true that in some cases the same species may be found in different environments.*

Navigation: 1. Drag and drop map into the report, choose spatial query option → next

**MSBI 2012 (SQL Server Integration Services)** → choose the dataset which has <sup>2008</sup> spatial

**New Map Layer**

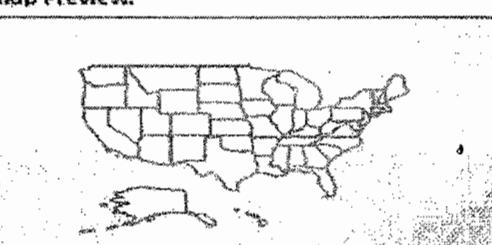
→ next → color analytical map → choose dataset → specify the below  
 Choose a source of spatial data options → Theme, field to visualize, color rule  
 Spatial data contains sets of coordinates that define map areas. display labels, data fields  
 → click finish and preview

- ① **Map gallery**  
 Use spatial data from a set of installed maps.
- ② **ESRI shapefile** Where can I get ESRI shapefiles?  
 Use a link to an ESRI shapefile (.shp)
- ③ **SQL Server spatial query**  
 Use a query that returns spatial data from a SQL Server database.

**Map Gallery:**

- ⊕ **USA**
  - ⊕ **States by County**
  - ⊕ **USA by State**
  - ⊕ **USA by State Exploded**
  - ⊕ **USA by State Inset**

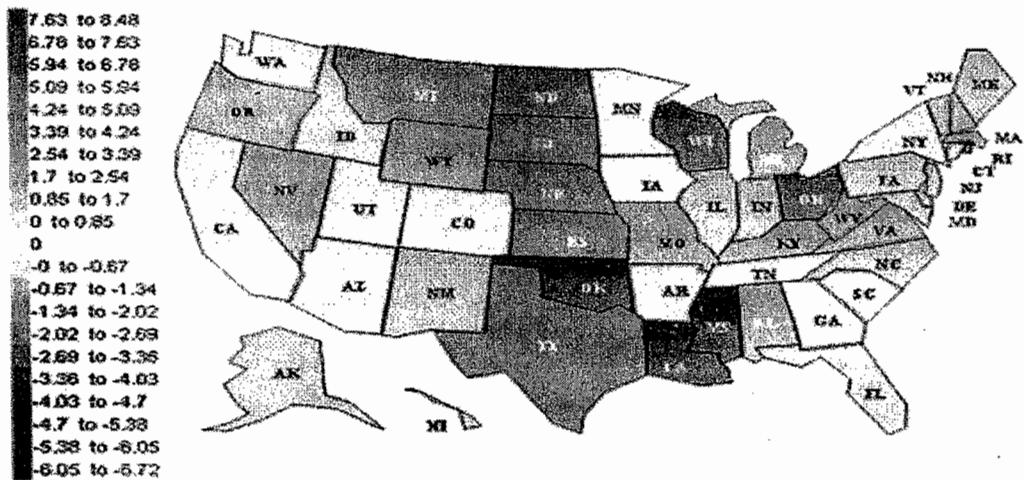
**Map Preview:**

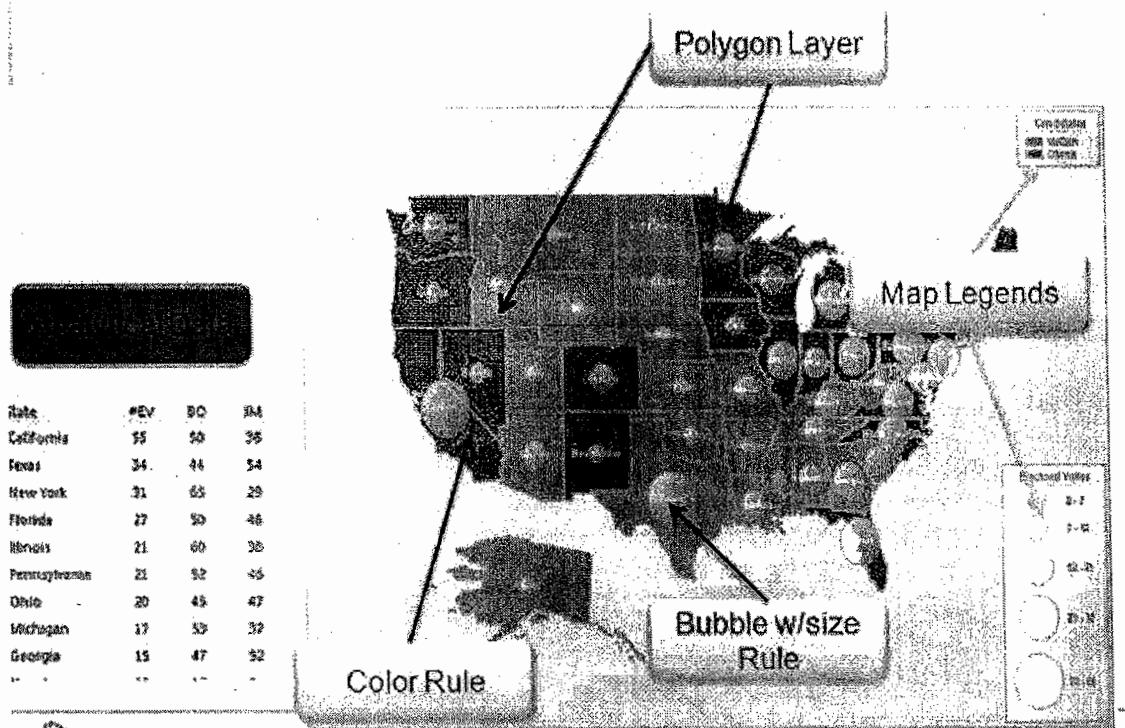


Note: To change fonts or settings goto polygon properties and change

[Help](#)    [Back](#)    [Next >](#)    [Cancel](#)

**GIS Map Chart Report**





# Vinay Tech House

## SUB REPORTS

1. Subreports to a report when you want to create a main report that is a container for multiple related reports. A sub report is a reference to another report.
2. When you add a sub report to the main report, you can specify parameters to pass to the sub report.
  - i) Create main report with the help of dept table and emp database
  - ii) Create sub report with the help of emp table and emp database.
  - \* iii) Make it sub report as parameterised report (Create deptno parameter).
  - iv) select parameter → parameter prop → select parameter visibility  
 Hidden

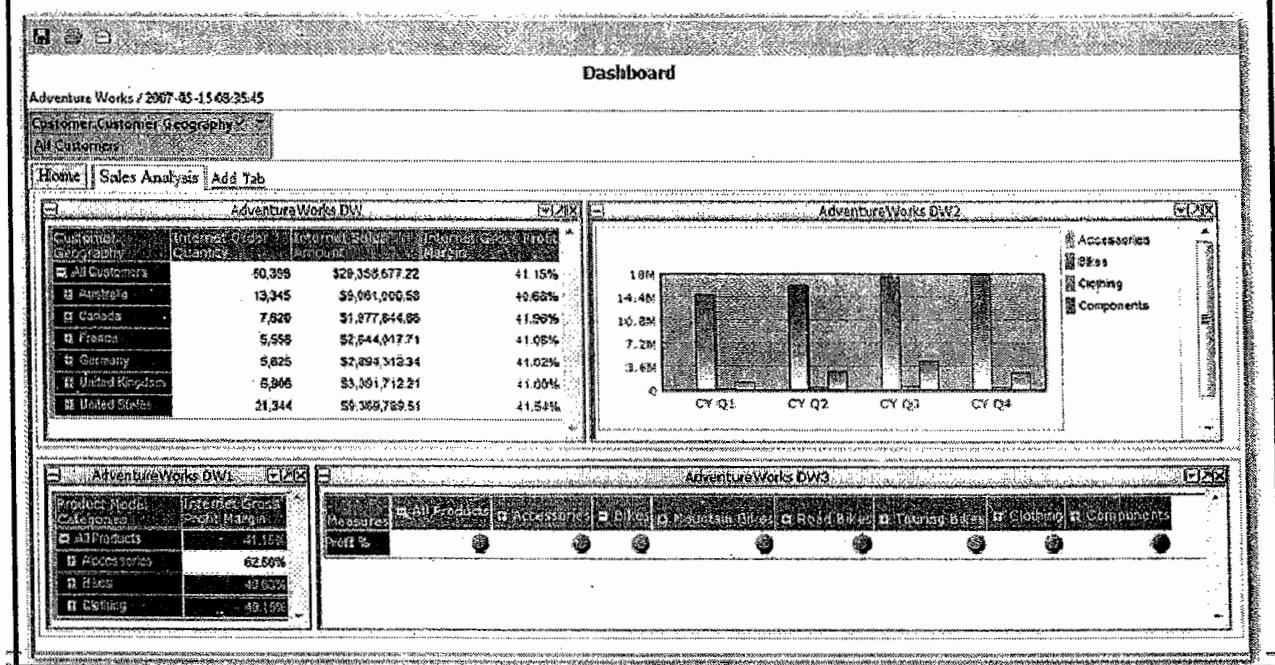
→ goto main report select entire data section → rc → insert row inside group below

- v) select all the cells → rc → merge cells. → within the cell → rc → insert subreport.
- vi) select the sub report → rc → sub report properties → General → use this report as a sub report  
SUB report

## INTEGRATED DASHBOARD REPORT

# Vinay Tech House

## Integrated Dashboard Report



## LOOKUP FUNCTIONS (Lookup, LookupSet, MultiLookup)

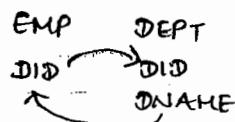
Introduced in 2008 R2 for two reasons.

- Multiple columns from multiple datasets into a single region.
- Lookup helps to get the required columns in the other datasets.

There are 3 lookup functions available.

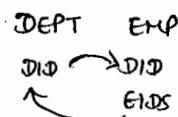
- Lookup
- LookupSet
- MultiLookup

FIRST: Lookup (1-1)



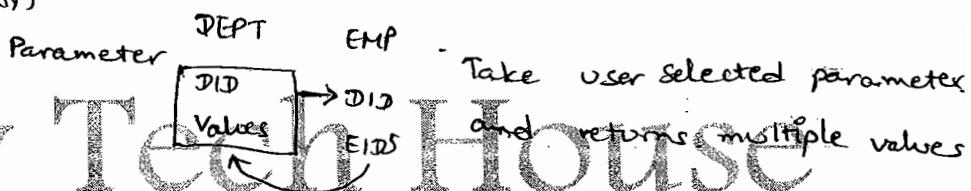
Matches single value and gets single value.

2nd: LookupSet (1-MANY)



single value match and returning multiple values as resultset.

3rd: MultiLookup (MANY-MANY)



Take user selected parameter and returns multiple values

COMMON SYNTAX:

LOOKUP / LOOKUPSET / MULTILOOKUP ( SOURCE MATCH COL , DEST MATCH COL , DEST RESULT COL , "DEST DATASET NAME" )

JOIN: Joins multiple values with the specified DELIMITER.

SPLIT: Split the values based on the DELIMITER.

Practices:

- CREATE EMP\_DST DATASET with this query -- select \* from EMP.
- CREATE DEPT\_DST DATASET with this query -- select \* from DEPT
- CREATE EMPI\_DST " " " " -- select \* from EMP\_NEL
- CREATE DEPTI\_DST " " " " -- select \* from DEPT1

Lookup Example:

	EID	ENAME	DID	
=	[EID]	[ENAME]	[DID]	<Expr>

= LOOKUP(Fields!DID.Value, Fields!DEPTID.Value, Fields!DNAME.value, "DST-DEPT")

LOOKUPSET:

DEPTID	DNAME	
= [DEPTID]	[DNAME]	<<Expr>>

\* = JOIN(LOOKUPSET (Fields! DEPTID.Value , fields! DID.Value , Fields! EID.Value, "DST\_EMP"), ",")

MULTI LOOKUP:

EID	DIDS	<<Expr>>
= [EID]	[DIDS]	<<Expr>>

= JOIN(MULTILOOKUP (SPLIT (Fields! DIDS.Value, ","), Fields! DNAME.Value, "DST\_DEPT"), ",") .

# Vinay Tech House

## Multi Lookup tables

EMP

eid(int)	ename(string)	dids(string)
1	vinay	10,20
2	madhu	30,20

DEPT

did(string)	dname(string)
10	it
20	hr
30	sales

## WORKING WITH REPORT MODEL

MSBI 2005, 2008: Model is mandatory for report builder operations.

MSBI 2008 R2: Model is optional. (shared dataset concept available).

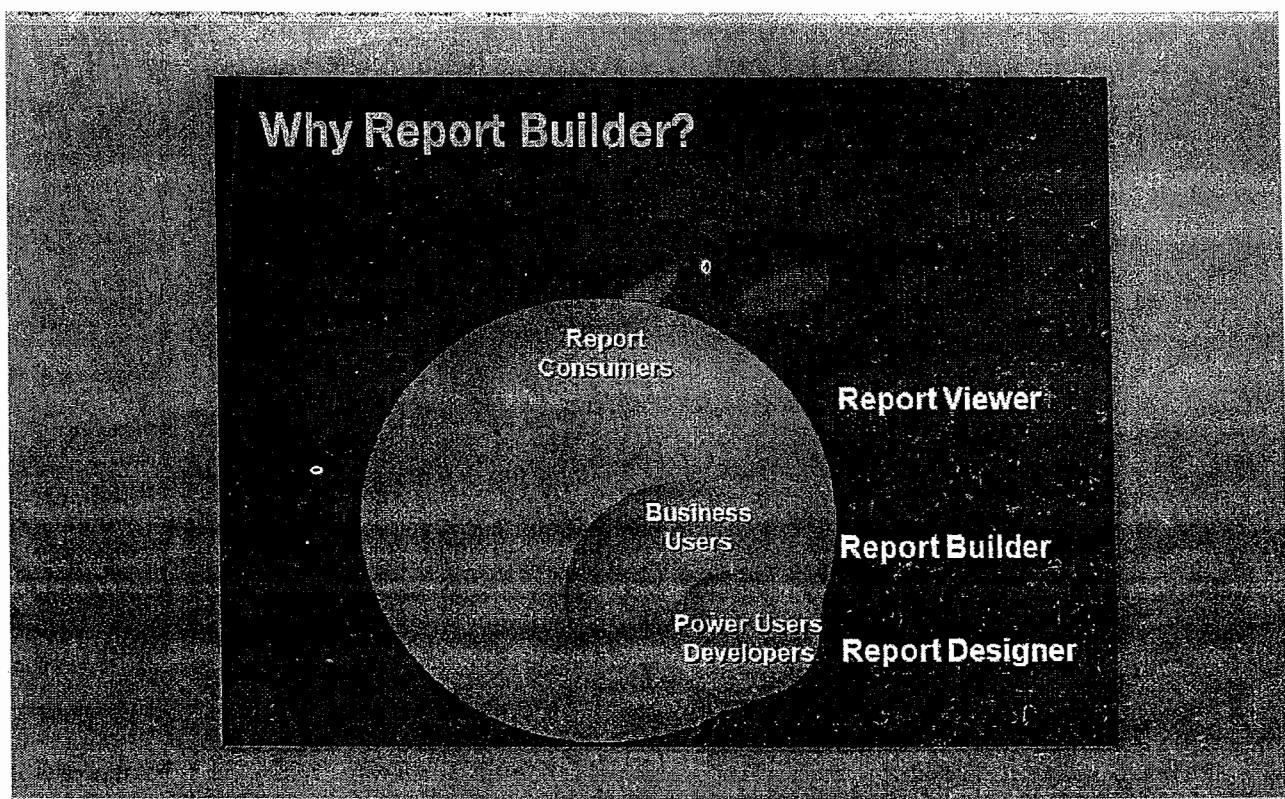
MSBI 2012: Model is removed.

Report model contains a set of tables and their attributes which are going to be across multiple adhoc reports. The report model statistics store under SMDL (Symantec model definition language.)

# Vinay Tech House

## WORKING WITH REPORT BUILDER

1. Report builder supports simple , Complex , dynamic and all reports.
2. Report builder supports security report processing options , report subscription options.
3. Report builder 1.0 → 2005 , 2.0 → 2008 , 3.0 → 2008 R2 , 2012.
4. If you save the report file in report builder it will available in report mgr.



### Report Builder is...

- a new ad hoc report design tool for SQL Server Reporting Services
- targeted at business users who want to find and share answers to interesting questions
- driven from a business model of the data so users do not need to understand the underlying data structures

**HOME**

Design Report

Run Report

Clipboard options

Paragraph Options

Border Options

Number Options

Font Options

Layout & Alignment

Merge--> Merge multiple cell values in to single cell

Split--> Split Single cell into multiple cells.

**INSERT MENU****Vinay Tech House**

The below report data regions and items supported.

- a) TABLE b) Matrix c) List d) Indicator e) Subreport
- f) Chart g) Maps h) Databasrs and Sparklines i) Line
- j) Textbox k)Image l)Footer m) header n) rectabgle
- o) Report Part etc...

Note:

Old report builders support only table,matrix and chart **reporing.**

**VIEW MENU**

1. Report Data : Displays parameters, Data sources, Data sets etc
2. Properties : Displays the properties of an object.
3. Grouping : Displays row and column groups.
4. Ruler : Takes ruler for a report.

**FILE MENU**

1. New : To create the new report.
2. Open : Opening report from report server directly.
3. Save,Saveas : Saving report directly in report server
4. Publish report parts : To publish report parts.

**REPORT PART**

1. Introduced in 2008R2 basically defined for to re-use a part of report which is well designed across multiple reports.
2. It helps to save the design time, Coding time in multiple reports.

Report part Navigation:

Creation: 1. Open report builder → Blank report → Create chart report

Saving: 2. Report builder → Click on ➡ publish report parts → publish all reports parts with default settings

Usage: 3. Report builder → blank report → Insert report parts → search on the Report part gallery

Modify: 4. RC on report → change chart type → select the req. template → Ok.

# Vinay Tech House

SQL server — 2008 R2 (MSBI)

Report part → Report Builder

Report Model (2012 They Removed → 2008R2 → Practical)

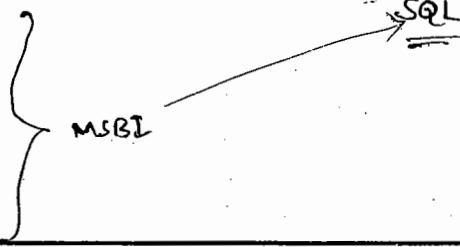
C → Functions

JAVA, C# → CLASS

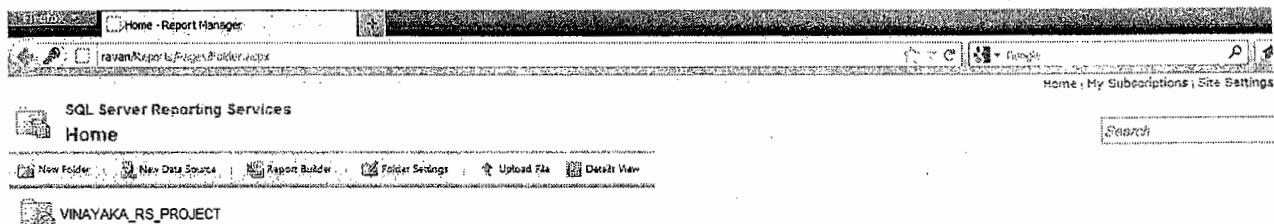
1. Syntax errors
2. Runtime errors
3. Design Time
4. Implementation time

\* SQL → STORED PROCEDURE \* \* \*

1. Syntax errors
2. Runtime errors
3. Hiding complex
4. Precompiled statements
5. RESIDES on servers
6. IN and OUT parameters



## WORKING WITH REPORT MANAGER



- This is the place where all the report operations take place.
- It is the client front end tool to access the managing the reports.
- Here we do generally
  - a) Creating Folders, Uploading Files, Showing details
  - b) Creating Caches, snapshot and maintaining the history
  - c) Providing a security
  - d) Creating subscriptions
  - e) Working with adhoc report etc. ....

Creating a new folder:

# Vinay Tech House

Creating a new Data Source: The reports can be executed with other environment settings. To do this multiple data sources we create, store the credentials permanently.

Monitoring Folder Settings It has 4 options. i) More ii) Delete iii) Security iv) My  
Here security is role based security, for a windows user or a group we assign  
respective responsibilities.

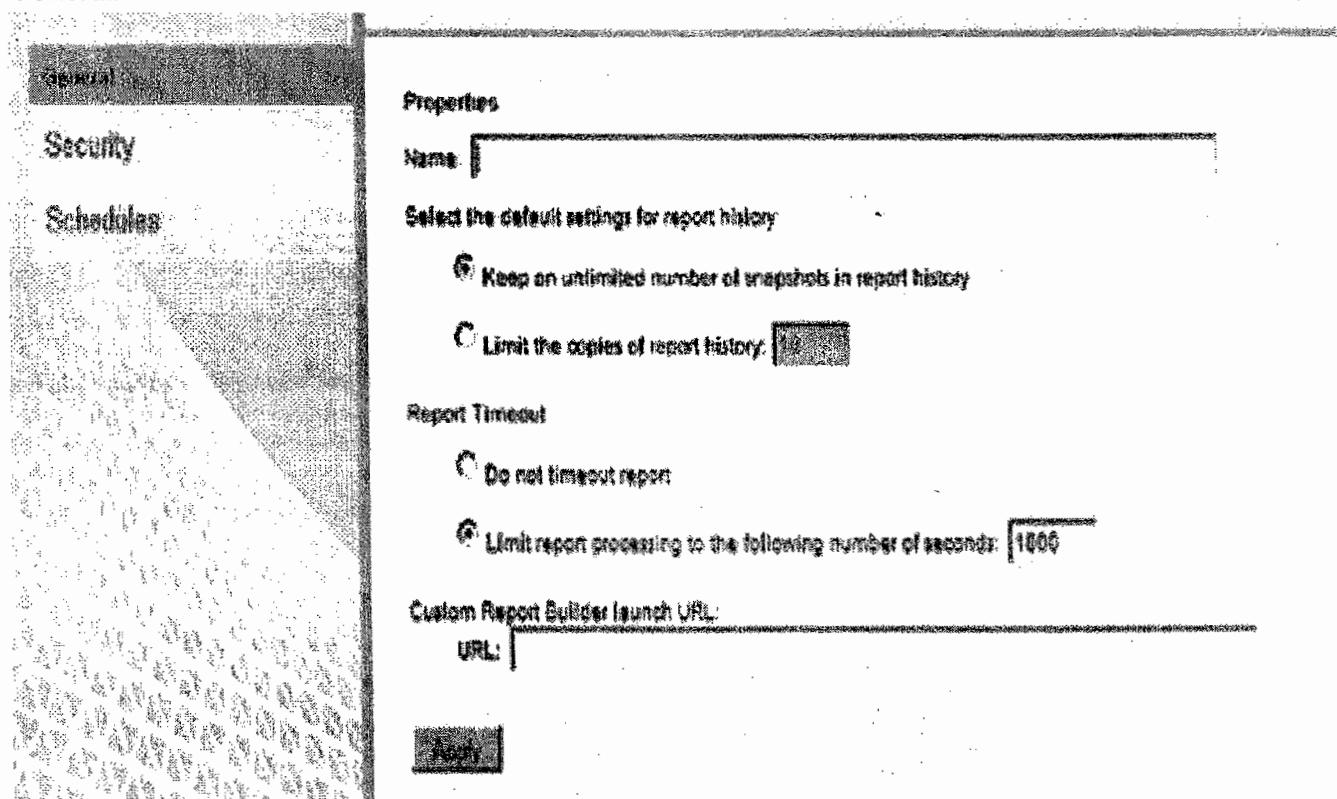
Upload File It helps to upload rdl file.

Details View There are 2 views. i) Details view ii) Tile view.

My Subscriptions:

Working with Site Settings:

General:



Security:

## Reporting Services Security

- Hierarchical Security Model
  - ▶ Folders can be used for logical organization
  - ▶ Items inherit permissions
- Security Layers
  - ▶ System-Level Role Definitions
  - ▶ Site-wide Security
  - ▶ Item-Level Role Definitions

## Managing Security

- Role-Based system
  - ▶ Roles are sets of permissions/capabilities
  - ▶ Users can be assigned to multiple roles
- Based on Windows Authentication
  - ▶ Provides for centralized security management
  - ▶ May use Active Directory users and groups
  - ▶ Other authentication can be developed

## Schedule

**Scheduling**

Use this page to create or modify a schedule.

Schedule name: **SC\_Shared\_Daily\_3am**

Schedule details

Define a schedule that runs on an hourly, daily, weekly, monthly, or one time basis.

All times are expressed in (GMT -07:00) Pacific Daylight Time.

Hour  
 Day  
 Week  
 Month  
 Once

**Daily Schedule**

On the following days:  
 Sun  Mon  Tue  Wed  Thu  Fri  Sat

Every weekday  
 Repeat after this number of days:

Start time:  :   A.M.  P.M.

Start and end dates

Specify the date to start and optionally end this schedule.

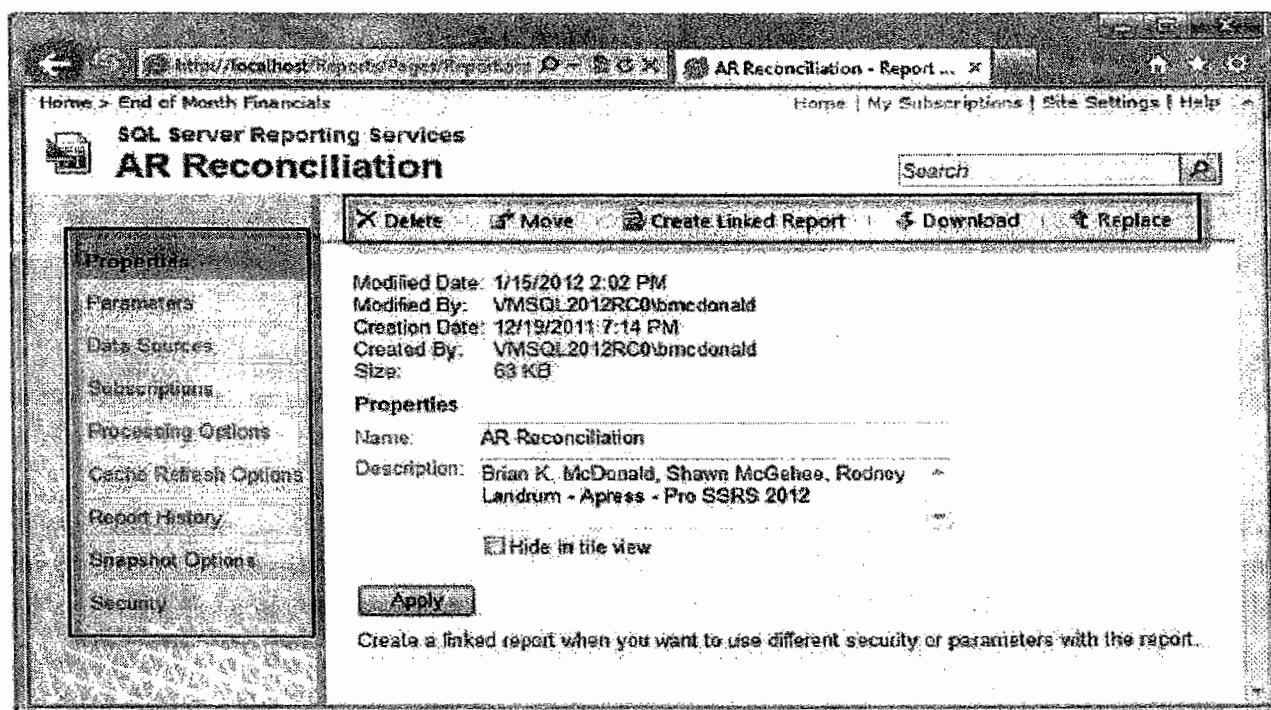
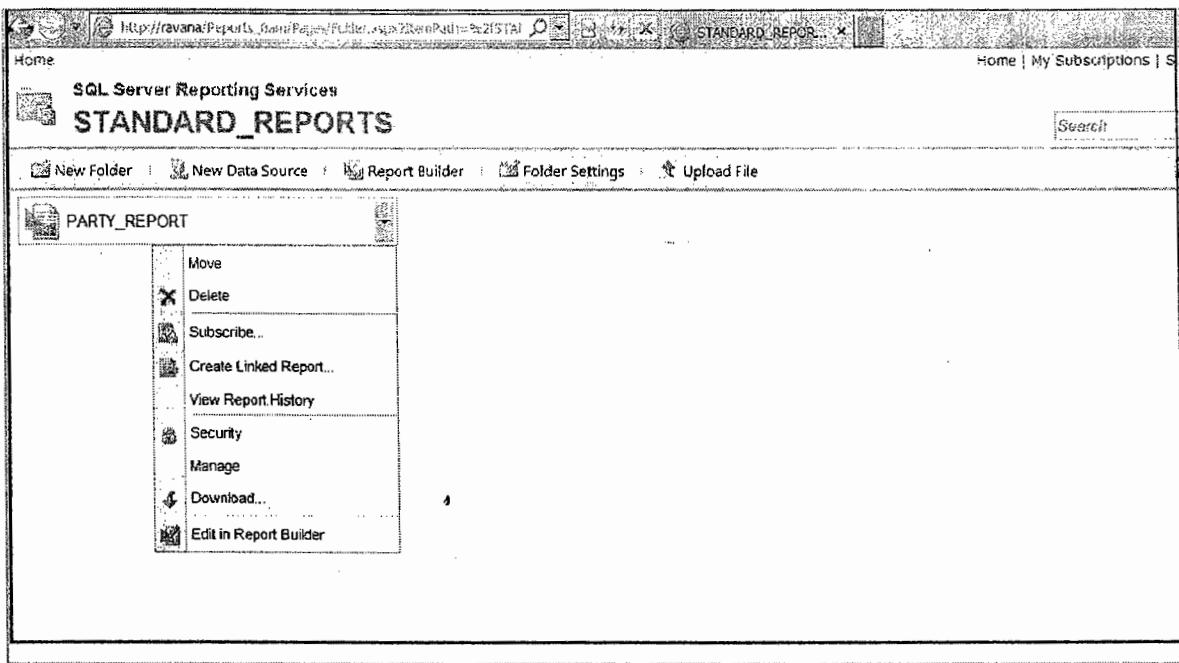
Begin running this schedule on:

Stop this schedule on:



Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12

## WORKING WITH INDIVIDUAL REPORT PROPERTIES



***Hide the report in the tile view:*** This can be useful if you don't want users with access to the SSRS 2012 server to know that certain reports exist or if you don't want them to see the detailed information about the report. Remember that this only hides the report in the tile view and not the detail view. You should use SSRS security properties when you need to prevent a user from seeing the contents of folders or running reports they shouldn't have access to.

Then, the menu going across the top of the screen allows you to:

**Delete the report:**

This gives you the ability to remove reports that are no longer needed.

**Move the report to another location on your report server:**

This allows you to organize the reports into folders for organizational and security purposes.

**Create a linked report:** This option allows you to create a linked report that is

based on the original report. This allows you to keep one copy of the report for deployment somewhere on the server, but have multiple copies available in other locations on the server without the need to redeploy them each time the report is changed.

**Download the RDL:** You can download the report RDL to a file that you can edit.

This can be useful when you want to make a minor modification to a report, such as changing the spelling of a word or modifying an expression. Keep in mind that this method only provides you access to the RDL file; you still have to use another program to edit the file and then you have to upload the modified file.

**Replace the RDL by uploading a new copy of the RDL file:**

If you've downloaded and edited the RDL file, then this option allows you to upload the edited copy.

**Make report parameter modifications:** From the Parameters section, you can change some options on the report parameters, such as default value, visibility, user prompting, and parameter display text. This ability to change parameters is convenient, since you do not have to download a report to make simple parameter changes.

**Make changes to the data source:** The Data Sources section allows you to change the data source used by the report. You can change to another data source located on the same server, or you can create a new custom data source from scratch.

**Manage Subscriptions:** The subscription sections allow you to manage subscriptions and data driven subscriptions of the report. You can create, delete, and modify your report subscriptions from this section.

**Processing Options:** This section allows you to configure data caching options for use when rendering reports. It also allows you to set up rendering of the report using a report snapshot and configure the report timeout properties.

**Cache Refresh Options:** You can set up a new cache refresh plan or make modifications to current plans from this section. These plans will create refresh jobs that update cache stores used for reports or shared data sources. This data will be used if your report is set up to render from cached data and not from live data.

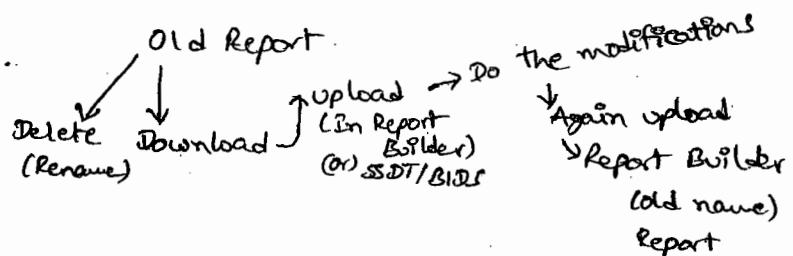
**Report History:** This section allows you to take snapshots of the report or to view historical snapshots of rendered reports. These can be created manually or by a snapshot schedule that is configurable in the next section.

**Snapshot Options:** From this section, you can manage snapshots and snapshot schedules. You can create a new snapshot schedule or set up a new snapshot on a shared schedule. You also have the ability to manage options for snapshot security and snapshot storage limits.

**Manage security settings:** You can manage individual security settings for a report from the Security section of the page. By default, a report inherits the parent folder's security settings, but you can break that inheritance (or restore it) from this menu.

## DOWNLOAD AND REPLACE REALTIME USAGE

1. To implement change requires on the report this options are useful.
2. Download helps to take the roll of the reports, and manipulate the report in SSDT (or) BI Studio by adding as existing item.
3. Replace is useful to replace with current report content with the modified report in the solution.



## LINKED REPORTS

# Vinay Tech House

### CREATING LINKED REPORT:

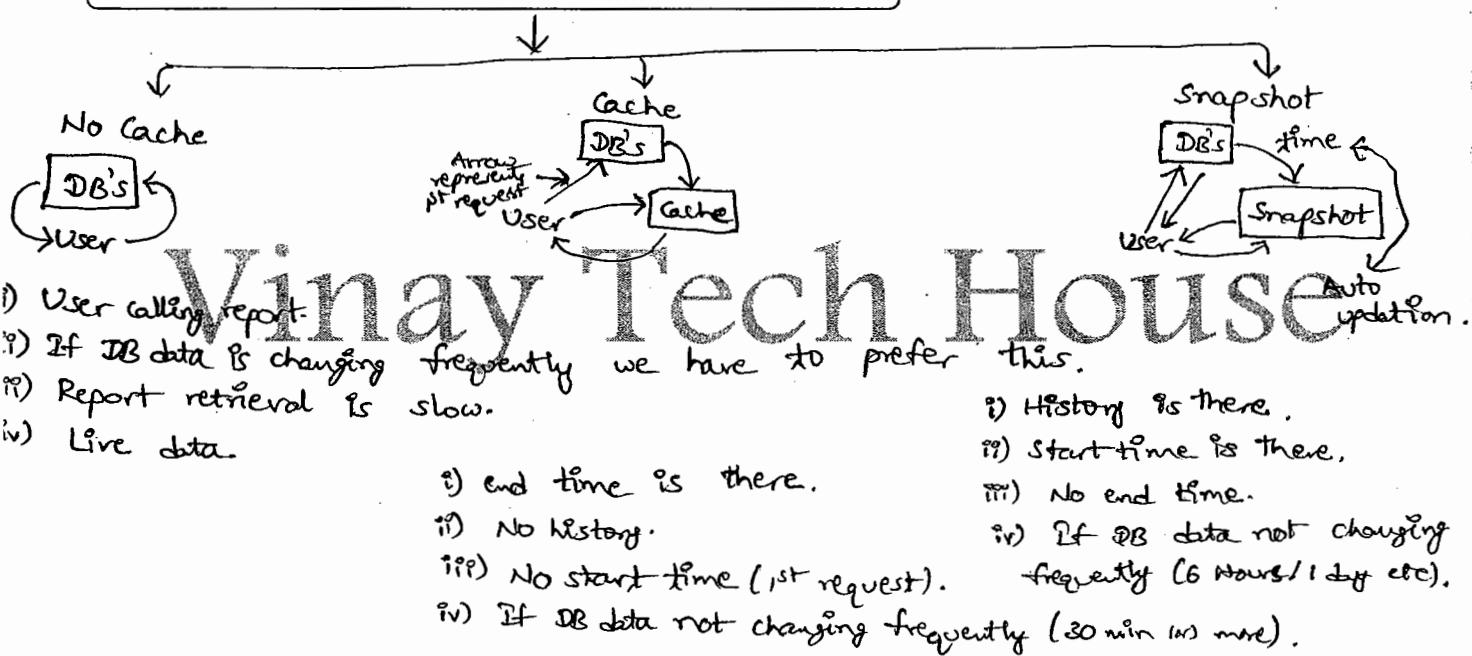
+

1. CREATE A PARAMETERIZED REPORT (ALL LOCATIONS) AND DEPLOY
2. GOTO MANAGER ,CREATE TWO FOLDERS (HYD\_DATA,MUM\_DATA)
3. GOTO THE REPORT AND DO FOR FIRST LINK (HYD\_DATA)
  - A) MANAGE->PARAMETERS->HAS DEFAULT SELECT AND SPECIFY HYD
  - B) GOTO MANGE PROPERTIES ->CLICK CREATE LINKED REPORT AND SPECIFY REPORT NAME AND SPECIFY CHANGE LOCATION TO HYD\_DATA FOLDE
4. DO THE SAME PROCESS FOR MUM LINK ALSO BY REPLACING HYD TO MUM IN ALL PLACES
5. GO TO HYD DATA AND MUM DATA FOLDERS AND OBSERVE RESUL

## Linked Reports

- Creates a "virtual report"
  - Uses the same report definition (.rdl) as the parent report, but with independent settings
- Purpose / Benefits
  - Can setup different sets of permissions
  - Can setup different sets of parameters

## REPORT PROCESSING OPTIONS



These are nothing but copy of the report.

Cache Instance	Snapshot
Here there is no start time. But end times is there.	There is a start time but there is no end time.
When the user requests a report a cache created & wait for expiration.	It is created at the specified time.
Small data report processing with frequent requests we go for it.	For long running queries & huge data reports we go for this.
Request first and cache is next.	Snapshot first and request next.
If the data is changing very frequently like 20 min. or 30 min. it is recommendable.	The data is not changing such as a longtime this is recommendable.
We cannot maintain history.	Supports support History Ex:- Sensex information not changes between 3pm to next day morning 9:10 am. So we can go for snapshot creation every day at 3:50pm. So that multiple users can connect the report & see the same content.

DB Name : ReportServer temp db

Table : Execution Cache

DB Name : Report Server

Table Name: Snapshot Data

## REPORT PROCESSING OPTIONS PRACTICALS

The screenshot shows the 'Processing Options' section of the PARTY\_REPORT report in the SQL Server Reporting Services interface. The left sidebar lists options like Properties, Date Sources, Subscriptions, Processing Options (which is selected), Cache Refresh Options, Report History, Snapshot Options, and Security. The main content area displays various processing options:

- Always run this report with the most recent data
- Do not cache temporary copies of this report
- Cache a temporary copy of the report. Expire copy of report after a number of minutes:
- Cache a temporary copy of the report. Expire copy of report on the following schedule:
  - Report-specific schedule:   
At 8:00 AM every Mon of every week, starting 11/15/2012
  - Shared schedule:
- Render this report from a report snapshot:
  - Use the following schedule to create report snapshots:
    - Report-specific schedule:   
At 8:00 AM every Mon of every week, starting 11/15/2012
    - Shared schedule:
  - Create a report snapshot when you click the Apply button on this page

# Vinay Tech House

## SUBSCRIPTION & TYPES

Subscription: Delivering the report in the specified time to the specified user in the req. delivering channel is called Subscription.

Delivery channel: a) Email b) Shared folders c) NULL delivery

Models: a) Push model - uses subscription method to deliver the report.

b) Pull model - Manually grabbing the report from server (or) manager.

c) stipulated time - daily weekly ... at a specified time

Subscription types: a) Standard - One user delivery with standard / fixed settings  
b) Data-driven - Multiuser delivery with variable settings.

→ When do we use NULL delivery subscription, what it is?

It will not deliver any report but used for cache refreshment

## Subscription Types

- Snapshot-Based Subscriptions
  - ▶ Notification is sent whenever a snapshot is created
- Schedule-Based Subscriptions
  - ▶ Uses a custom schedule (e.g., daily, monthly, etc.)
  - ▶ Can have start and stop dates
- Data-Driven Subscriptions
  - ▶ Report recipients are defined by a query
  - ▶ Table and query must be created manually
  - ▶ Useful when managing large or very dynamic lists of recipients

**STANDARD SUBSCRIPTION (EMAIL)**

2. SMTP server is required to delivery.

1. To implement this email setting should be specified at report mgr.

<input type="checkbox"/>	<input type="checkbox"/>	Description	Trigger	Last Run	Status
<input type="checkbox"/>	<input type="checkbox"/>	Sales Order delivery	Timed Subscription	3/2/2012 11:20 AM	Done: 4 processed of 4 total; 0 errors.

Subscription: Patient Recert...

Home | My Subscriptions | Site Settings | Help

## Subscription: Patient Recertification Listing

Report Delivery Options

Specify options for report delivery.

Delivered by: E-Mail

To: Brian\_K\_McDonald@sqlbiggeek.com

Scc:

Bcc:

(Use () to separate multiple e-mail addresses.)

Reply-To:

Subject: @ReportName was executed at @ExecutionTime

Include Report Render Format: MHTML (web archive)

Include Link

Priority: Normal

Comment:

Subscription Processing Options

Specify options for subscription processing.

Run the subscription:

When the scheduled report run is complete.

At 8:00 AM every Mon of every week, starting 1/20/2012

On a shared schedule: End of Month Financials

At 12:01 AM on day(s) 1 of every month, starting 12/19/2011

Report Parameter Values

Specify the report parameter values to use with this subscription.

Branch:

Freedom Center  Use Default

## STANDARD SUBSCRIPTION (FILE SHARE)

Here we render the Report to a specified shared folder.

- Practices:
1. Specify (or) Create a shared folder (right click folder share with - type everyone - Add).
  2. Goto report mgr goto individual properties by clicking manage option and do the below settings.
    - a) Specify credentials to the data source
    - b) subscription - new subscription - windows file share

Filename : PARTY-REPORT

Path : \\vinaytech-pelarc

RenderFormat: Word

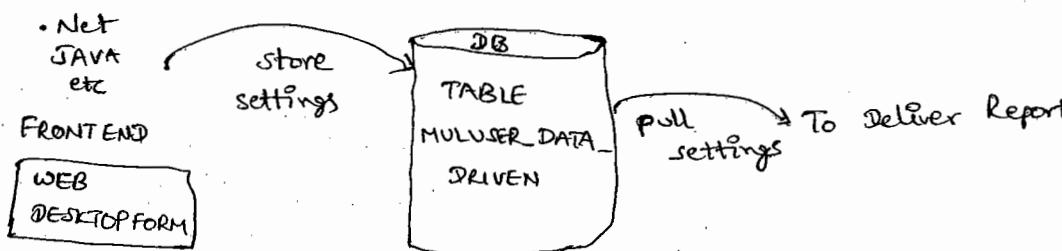
Username : vinaytech → windows credentials folder  
to connect to shared folder  
password :

# Vinay Tech House

## DATA DRIVEN SUBSCRIPTION

1. To implement this we need

- one data base and a table. (This table contains the req. settings to deliver.
2. In Real time this table data is populated from front end applications



Practices: Create a table like below.,

Column Name	Data Type
FILENAME	VARCHAR(50)
PATH	"
RENDER_FORMAT	"
WRITE_MODE	"
FILE_EXT	"
USERNAME	"
PASSWORD	"

click next → specify schedule → finish  
 Monitor the subscription for last run.  
**MSBI 2012 (SQL Server Integration Services)**

222

## 2. Arc and Arc1 folders.

PARTY NAME	FILENAME	PATH	RENDER FORM	WRITE MODE	FILE EXT	USERNAME	PWDP
1	REPORT_DATA	\\\vinaytech\pc\arc	Excel	None	True	VINAYTECH	NULL
2	REPORT_DATA1	" arc1	PPF	"	True	VINAYTECH	NULL

articles:

1. Similar to standard subscription

2. Click data-driven subscriptions, specify description, windows fileshare option, specify for the subscription only and click next → specify datasource settings.

DATA SOURCE = \\\vinaytech\instance → server name  
 Initial catalog = DB\_MSBI; → database name

Specify windows credentials → Next.

Write a query select \* from msuser\_data\_driven → click validate → Next,  
 Assign table columns to every settings like below., FILENAME

- Get the value from DB : FILENAME
- Get the value from DB : PATH

**SECURITY**

# Vinay Tech House

New Role Assignment - Report Manager - Windows Internet Explorer

Home | My Subscriptions | Search

SQL Server Reporting Services

New Role Assignment

Use this page to define role-based security for PARTY\_REPORT.

Group or user name: DEVELOPER

Select one or more roles to assign to the group or user.

Role	Description
<input type="checkbox"/> Browser	May view folders, reports and subscribe to reports.
<input type="checkbox"/> Content Manager	May manage content in the Report Server. This includes folders, reports and resources.
<input checked="" type="checkbox"/> My Reports	May publish reports and linked reports, manage folders, reports and resources in a users My Reports folder.
<input type="checkbox"/> Publisher	May publish reports and linked reports to the Report Server.
<input type="checkbox"/> Report Builder	May view report definitions.

OK Cancel

**CREATE A SAMPLE REPORT IN POWER VIEW**

1. This concept introduce in 2012, specially available in SharePoint server.
2. Power view has many additional features such as project cards, tiles, cards etc.
3. Power view reports are helpful for team and personal BI operations.

# Vinay Tech House

## PERFORMANCE TUNING & OPTIMIZATION METHODS

FAQ's

1. How do we optimize a report.
2. If report is long time to run (a) occupying more memory how do we eliminate that?

For above 2 questions the solution is performance tuning.

Identify eliminating bottlenecks <sup>weak</sup> is a process of performance tuning.

bottlenecks: It is not an error but causes delay in performance. There are many ways to identify bottlenecks eliminate.

The best way is opening execution log views. (These views poll data from ReportServer).

WORKING WITH EXECUTION LOG VIEWS: Read the fields information in the log views and take solution accordingly.

Time retrieving these provide execution statistics of report at various levels.

→ by row These provide execution statistics of report at various levels.  
→ byte/row There are three types of Logs available (for detail description refer to page: 81).

To provide the settings for ExecutionLog:

ssms->objectexplorer->rs->properties->  Enable the Report ExecutionLogging  
source level, source DB, query tuning (or)  Remove the entries older than the no of Days [3]  
procedure/view tuning is mandatory.

Queries to Retrieve Log Info:

a) Select \* from ExecutionLog /ExecutionLog2 /ExecutionLog3

ExecutionLog2: This view is a good starting point from which to analyze your current workloads and understand its dataset size, performance and complexing characteristics.

It has a column called AdditionalInfo which contains the information related to the size of memory pressure responding data structures.

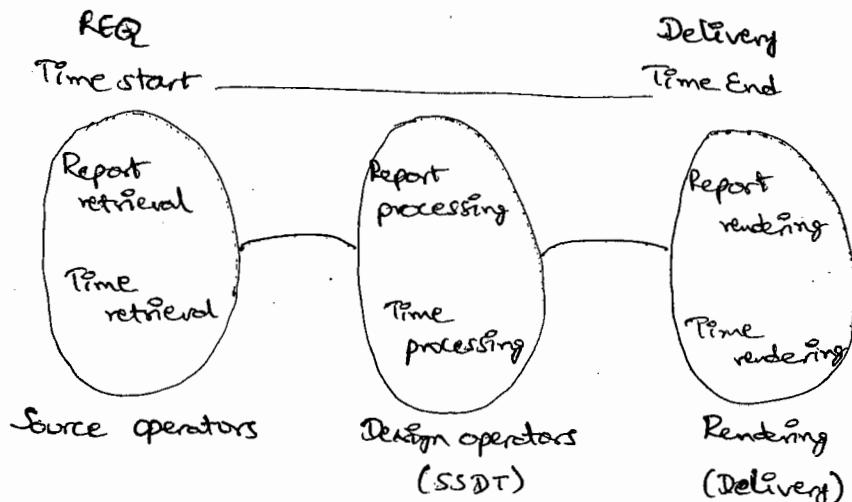
ExecutionLog is for ObjectLevel, ExecutionLog2 is for EventLevel

Tips: i) Eliminate Cross join, Correlated Subqueries.

ii) Eliminate having clause and use group by.

iii) Eliminate IN usage, if possible used exist clause.

iv) Eliminate ambiguous conditions and inequality range conditions.



v) Take only requires rows and columns.

vi) Time processing issue: Tips: Generally at report design level we will find the

i) Take only required items and regions.

ii) Do not hide any objects.

iii) Write simple calculations.

iv) Reduce the no. of Exp. in the report. (Too many exp. will spoil performance).

v) If really required take custom code.

vi) Do not take unnecessary columns in the dataset and do not keep unnecessary datasets in report level.

Report rendering issues: Generally at report access and delivery level we found these issues.

a) Use cache / snapshot for faster report retrieval.

b) Use NULL delivery option for cache refreshment.

c) Use subscriptions for better delivery.

d) Avoid linking to reports.

e) Use report builder for simple and faster reporting.

System level issues: 1. There are no issues in the above all levels defined by the problem should be at system level.

2. DBA's and other senior resources will participate to eliminate system bottlenecks.

# Vinay Tech House

**INDUSTRY BEST PRACTICES & METHODS**

TIPS

1. Use procedures or views for datasets for continuously to run without effecting underline datastructure change.
2. Use calculations in the report when you don't have access in the source level.
3. If you plan to use existing dataset rather than starting a new report copy existing report and delete unnecessary regions from it.
4. Properties that can be format with 4 values (2,2,2,2) always read from left to right, top to bottom. 2 2 2 2  
left right top bottom
5. List report is very powerful to use other regions inside.
6. In matrix report if you want some columns to appear before the columns grouping ~~for~~ property in specify an integer value in the groups before row header property.
7. If the report is not showing any data specify no rows property with a textual message.
8. Create Separate datasets from filters.
9. Parameters are used to dynamically control the formats and data in the datasets.
10. Filter gets full data in the datasets and then filter. So we need to use appropriate filters.

Eg: Filter data into one region, full data into another region.

11. It is recommended take underline for actionable items.
12. To disable pagenations (displaying in multiple pages) use interactive hide interactive with settings to zero ' value.

**SSRS INTERVIEW Q & A**

**Vinay Tech House**

## PERFORMANCE EXECUTION AUDITING & ANALYSIS (Copied)

SSRS provides a built-in logging feature that captures several key pieces of information.

**This information is useful in two ways:**

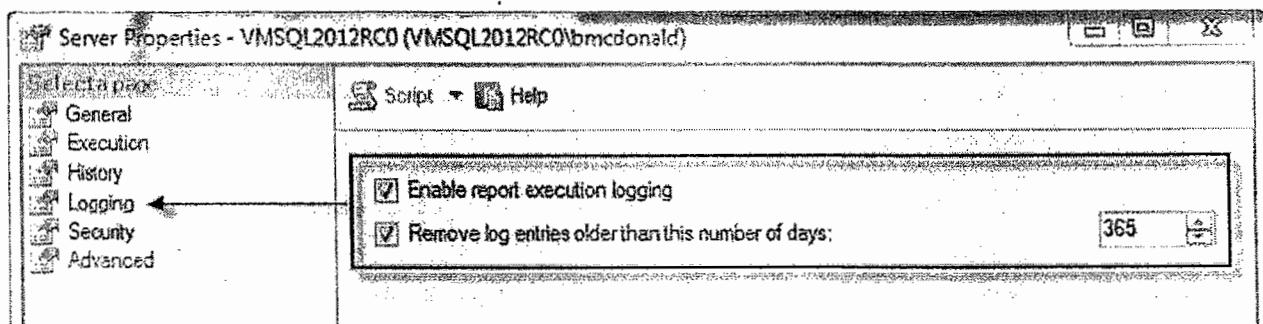
- You can capture performance information about the reports, such as the processing duration and record count.
- You can capture security information, such as who executed the report and whether or not they were successful.

**The first goal** in the following sections is to set up and extend the built-in logging functionality of SSRS using tools provided in the SSRS installation. You'll need to log all activity so that you can pinpoint the reports and users who are most impacting the server. We have created a custom SSRS report, Report Execution Log, which will deliver the logging statistics to administrators and contain dynamic column groupings based on a report parameter. We'll show how you can use this report for your SSRS deployment.

**The second goal** in the following sections is to show how to perform benchmarking tests on the SSRS servers in our test Web farm to ensure there won't be any unexpected performance problems when SSRS is deployed to a production environment. We'll show how to work with a Web application stress test utility called Application Center Test (ACT) to gauge performance.

### Configuring SSRS Logging

Getting to the execution log information in SSRS is a fairly straightforward procedure. It consists of a main table in the SSRS database called, appropriately enough, `ExecutionLogStorage`. When SSRS is installed by default, execution logging is enabled and set to maintain 60 days of logging. After 60 days, the log entries older than 60 days are removed from the table automatically. However, you do have the option to change the duration by connecting to the instance of SSRS using SSMS, then right clicking the server and selecting Properties. After the Server Properties windows comes up, navigate to the Logging tab. Figure 10-24 shows the settings if you wanted to change the number of days to hold 365 days rather than the default 60 days. With the execution details being stored and being that one of the aims is to build a custom SSRS report to deliver report execution information to administrators, you'll need to be able to query the log data. Fortunately for us, Microsoft has created three views that utilize the execution details to make it easy to determine information such as the user executing the report, duration a report took to process or render and even if it was executed interactively or via a subscription. The three views in the `ReportServer` database are `ExecutionLog`, `ExecutionLog2`, and `ExecutionLog3`. We will utilize these views as a basis for our administrative report.



## Transforming the ExecutionLog Table

Setting up SSRS to transform the logging data is a simple step-by-step procedure. Over the last several years, we have created many solutions to transfer and store the execution data into a historical database so that we do not lose any of the execution details when SSRS removes it from the database. One such solution is to use a SQL Server Integration Services (SSIS) package to transfer the execution history on a daily basis. As part of the Pro\_SSRS solution, we have included a SSIS project with a single package called Pro\_SSRS\_Load\_ExecutionLog.dtsx.

Furthermore, as part of the download is the Pro\_SSRSExecutionLog database. You will need to have this database restored in order for the Pro\_SSRS\_Load\_ExecutionLog package and the reports created in this chapter to run. See the ReadMe.txt file for detailed instructions on how to restore the Pro\_SSRSExecutionLog database. You can download the Pro\_SSRS project from the Source Code/Download section of the Apress Web site ([www.apress.com](http://www.apress.com)). The Pro\_SSRS\_Load\_ExecutionLog package is pretty straightforward. At the top of the package, we just check for the existence of the database and if it does not exist, it branches to a statement that creates the database, a table for archiving the Execution log and loads it with all records up to yesterday. If the database already exists and the records are not already present in the database, then all of those records are pulled into the archival database from the ReportServer database. Upon scheduling this package on a nightly basis, it will get the data for the prior day. Figure 10-25 shows the package results after completing a day's load when run interactively in the development environment.

**Tip** Note that the package needs to be executed regularly to keep the transformed log data current. In our situation, we created a scheduled job for this purpose that runs the SSIS package every evening.

Microsoft provides a set of sample reports that you can use with the ExecutionLog database. These reports are included in the SQL Server 2008 installation in the following location: Samples\Reporting Services\Report Samples\Server Management Sample Reports\Execution Log Sample Reports. The sample reports are useful for giving administrators information, such as report execution by user and report size, among other things. You will find three SSRS execution log sample reports in all. One of these, Execution Summary, is shown in Figure 10-26.

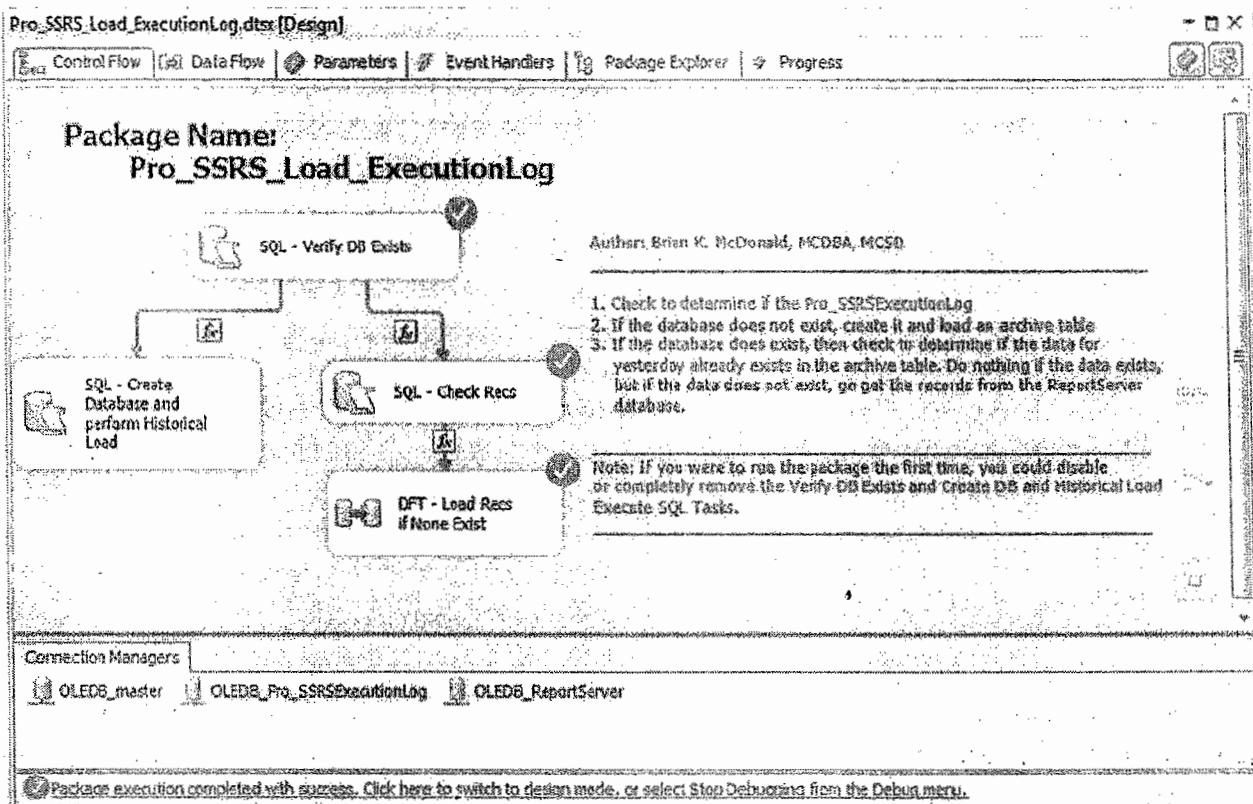


Figure 10-25. Pro\_SSRS\_Load\_ExecutionLog SSIS package

**Vinay Tech House**  
 In prior releases of SSRS, Microsoft has provided a set of sample reports that you can use to report data from the ReportServer database in conjunction with an SSIS package that can be used to transfer the data. However, as of today, no new version has been created for the 2012 release. As such, we have created a sample Execution Summary report that gives us a few meaningful metrics that our administrators want to keep an eye on. Some of those metrics include total number of executions, successful executions, failed executions, and number of executions by day number of the month.

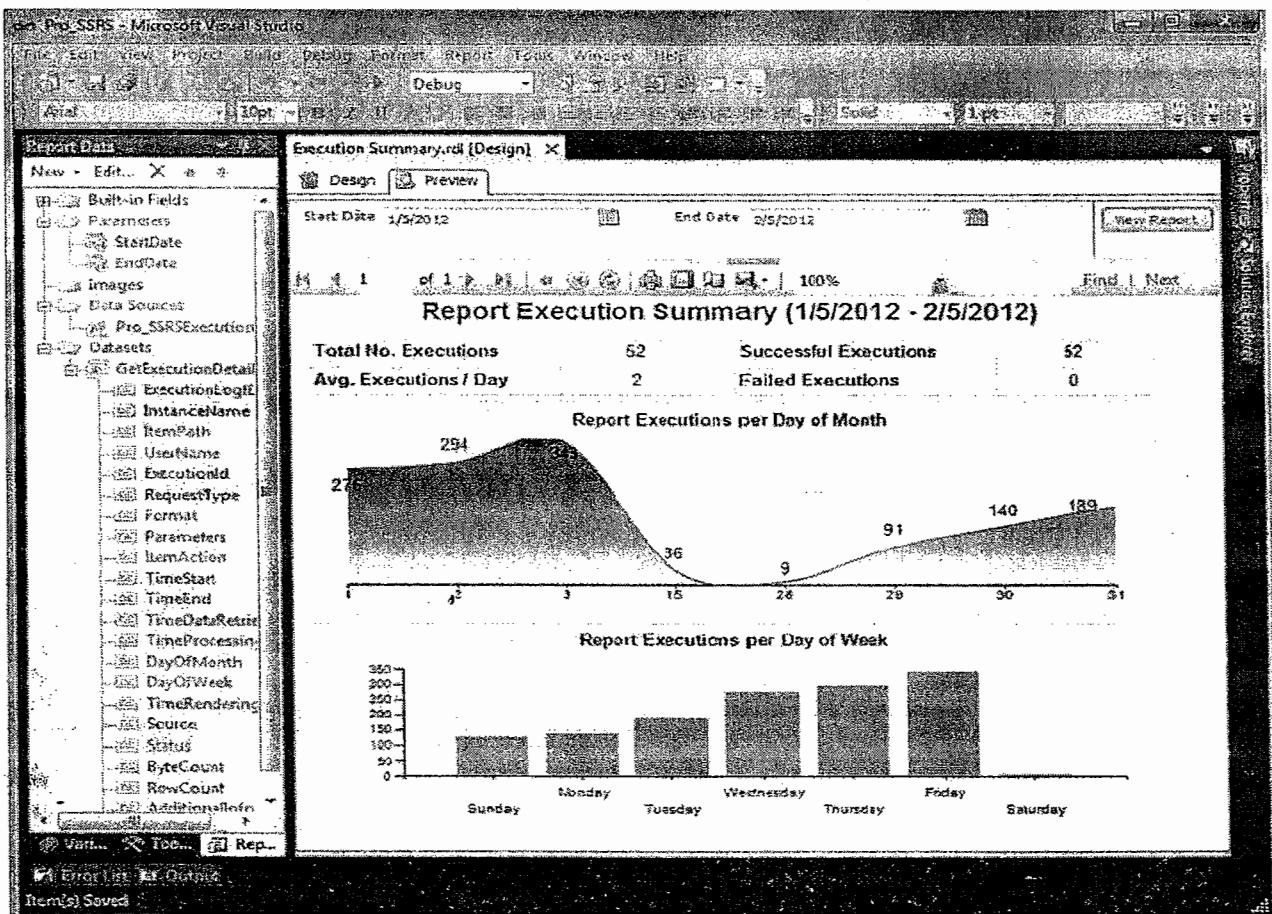


Figure 10-26. Execution Summary sample report

## Designing the Log Report

We knew we would need another report that contained all the execution log information and was easy for administrators to analyze. Thus, we created a matrix-style report, called Report Execution Log, using data from a single query. To measure performance from information contained in the execution log, you need several statistics:

- **Total time to retrieve the data:** How long did it take to retrieve data?
- **Total time to process:** How long did the report take to process?
- **Total time to render:** How long did the report take to render?
- **Byte count:** How many bytes are in the report?
- **Row count:** How many rows of data are in the report?

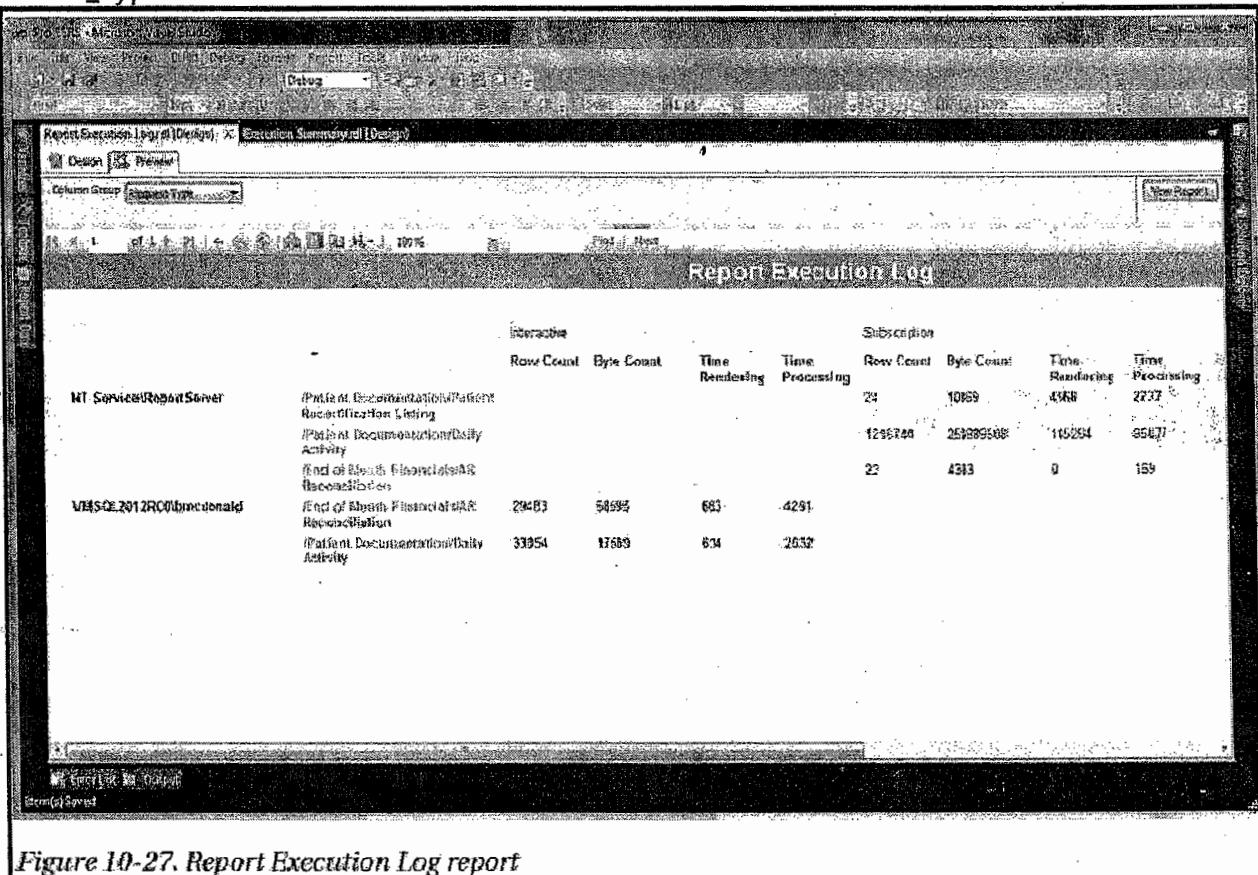
In addition, knowing when the report was executed is also useful. For the row groupings in the matrix, we want to see what report was executed, who ran the report, and from which client machine the report was run. For the column group, we want to have two possible selections: rendering format, such as HTML 4.0, Word or PDF, or source types, meaning how SSRS generated the report (live, cache, or snapshot, for example). Source\_Type is an important field to monitor because how SSRS generates reports directly impacts performance. Generating reports from a cached copy or a snapshot, which are both pre-processed copies of reports, is a

performance benefit. If SSRS is always generating live or on demand reports for users, performance may suffer.

To accomplish the dynamic column groupings in the matrix, you'll use a parameter called Column\_Group that takes the values of the field names in the query, Format or Source\_Type. You'll use a default value of Format so that the report is automatically rendered when previewed. Both the column grouping and heading values will use the following expression to make the column dynamic based on the parameter:

=Fields(Parameter!Column\_Group.Value).Value

When the report is rendered, as you can see in Figure 10-27, it will default to the Format field, but you can change it dynamically by changing the parameter drop-down selection to Source\_Type.



The screenshot shows the 'Report Execution Log' window in the SQL Server Reporting Services interface. The window title is 'Report Execution Log [ReportExecutionSummary1] [Design]'. It displays a table with two main sections: 'Interactive' and 'Subscription'. The 'Interactive' section shows data for two users: 'NT Service\ReportServer' and 'VINESO2012RC01\mcdonald'. The 'Subscription' section shows data for the same users. The table includes columns for Row Count, Byte Count, Time Rendering, and Time Processing.

		Interactive				Subscription			
		Row Count	Byte Count	Time Rendering	Time Processing	Row Count	Byte Count	Time Rendering	Time Processing
NT Service\ReportServer	Patient Documentation\Patient Recertification Listing	24	1089	4368	2237				
	Patient Documentation\Daily Activity	1295244	25989508	115284	6547				
VINESO2012RC01\mcdonald	Patient Documentation\Daily Activity	23	4313	0	169				
	Patient Documentation\Patient Recertification	29483	58592	683	4291				
	Patient Documentation\Daily Activity	33954	17689	634	2632				

Figure 10-27. Report Execution Log report

## Monitoring Performance

Of course, no one wants to experience the frustration of building a solid reporting solution in a development environment only to find out that, when deployed to the masses, it can't hold up under the strain. Generally, it's a best practice to put a simulated load on your servers to gain a better understanding of how the systems will function. As well, when you roll out a full solution, it's a common practice to roll out several pieces at a time to a limited number of users. That is what we've done in our online models.

The strategy for rolling out should also include a plan for which reports will be available on-demand versus which ones will be provided via report snapshots or subscriptions, as you've

done up to this point in the chapter. Combining a strategy of peak and off-peak report processing will greatly improve performance. Another consideration for performance with SSRS lies in splitting the load of SSRS Web services and database services. That is, if the entire SSRS installation resides on the same system, this could negatively impact performance.

In this section, we'll show the results of a stress test that we ran accessing two report server instances on two separate servers, RS05 and HWC04. Many tools, such as SSRS, are available for stress testing Web applications; fortunately, the Ultimate edition of Visual Studio 2010 has a Web stress-test tool built in that we used to perform a simulated load on the two servers with up to 250 virtual users out of the box.

We'll also show you how to use rsconfig to join an SSRS server to a Web farm to see how offloading resources to another system will enhance performance. The Reporting Services Configuration Manager can also be used for this purpose.

We began by running a simulated load of 15 users, all executing a single report against RS05. We quickly assessed the performance impact by monitoring the server with Task Manager. Fortunately, when running the simulated load against our Reporting Services server, it didn't take that high of a performance hit. However, in some instances you may get results that max out the CPU. In those instances, you are likely to find that the individual processes that are taxing the processor are SQL Server and the ReportingServicesService.exe. Figure 10-28 shows spikes of our CPU usage jumping between 25 and 85 percent as the test was running.

We knew our test SSRS server was a four processor system with more than 3GB of RAM. In the case where your CPU usage is getting maxed out, this tells us that it is the CPU that is the bottleneck. Because our production deployment of SSRS would not mirror the setup of RS05 (in other words, the production server would be a high-end system with at least 32GB of RAM), we could take that into consideration.

However, one other factor would have a substantial impact on the difference in performance between the production and test environments. In the test environment, the SSRS service and SQL Server were on the same system, RS05. What if we configured the SSRS service to use a remote SQL Server instance for its database? Any performance degradation caused by accessing the ReportServer database over the network instead of a local database would be negligible if the CPU utilization percentage dropped down to a more manageable number.

If you have two SSRS servers, then moving an SSRS server from one instance to another is simple. We had two SSRS servers in the test environment, RS05 and HWC04, so the move was easy enough. To instruct the SSRS service on RS05 to use the SSRS databases on HWC04, we used the command-line utility rsconfig. The rsconfig command or Reporting Services Configuration Manager is required when first joining one SSRS Web service to a Web farm that uses the same ReportServer database.

The syntax for the rsconfig command is as follows:

**rsconfig -c -s HWC04 -d ReportServer -a SQL -u username -p password**

With RS05 using the remote SQL Server database, we initiated another test to see whether the CPU utilization improved. The CPU utilization improved substantially and was now under the 60 percent average.