

Incremental Data Loads (INSERT only)...

As QlikView applications are developed to analyse more and more data, your organisation may want to incrementally load transactions. This means that QlikView will only load (from your source data), the 'new' data or latest transactions.

To enable an incremental load you have to utilise QVD files. The process for incremental loads is below.

1. Create your initial historical load from your data sources. This will create your base QVD that will be "added to" every time you reload your application. This QVD file will contain all of your transactions from history.
2. Define your key field with which you can determine the latest (new) records. This may be a date field or a sequential key or invoice number.
3. Find the latest or maximum value to pass to your SQL statement or subsequent reload.
4. Reload your data from your source using the passed value to restrict the records to load.
5. Add the 'new' data to your existing QVD/historical data
6. Store your updated QVD.

The Script below shows you a simple incremental load. This will carry out the basic function of inserting/adding new data. For more complex incremental loads such as UPDATE and DELETE. Please refer to the reference manual.

```
//Find the maximum id or latest date etc to pass to the variable and subsequently the
where clause of the data load
MaxKeyLoad:
LOAD Max(PrimaryKey) as MaxID
FROM MyData.qvd (qvd);

//Peek into the MaxHeadLoad to get the latest value, in this case the MAX load above will
//only bring back 1 row so the 0 in the variable tells it to bring the first row.
Let MaxID = peek('MaxID',0,MaxKeyLoad);

//Connect to your data source
CONNECT TO [Provider=OraOLEDB.Oracle.1;Persist Security Info=False];

//Run the data load to pull data based on the variable to find the latest / newest row
MyData:
LOAD PrimaryKey,
Column1,
Column2,
Column3,
Column4;
SQL SELECT *
FROM DATABASE.MYDATA
Where PrimaryKey > $(MaxID);

//Append the new updated data to the existing historical qvd file
CONCATENATE
LOAD * FROM MyData.qvd (qvd);

//Store the new larger qvd file
Store MyData into MyData.qvd;
```

