INTEGRATING HYPERION ESSBASE AND OBIA – A REAL WORLD CASE STUDY

Abhinav Banerjee, IBM

INTEGRATING HYPERION ESSBASE AND OBIA

This session will showcase a real world case study involving Hyperion / Essbase and Oracle Business Intelligence Applications (OBIA). We will look at leveraging the best of both the platforms to take advantage of Multidimensional capabilities of Essbase and pre-built data warehousing and BI capabilities of OBIA. We will look at the scenario where the customer manages account hierarchy and aggregated data in Essbase and the granular level data or detailed transactions in the Oracle Data Warehouse (OBAW). This is an ideal setting for drill-thru reporting. Often companies may want to maintain their complex account and organizational hierarchies in Essbase rather than in ERP systems like E-Business Suite.

There might be situations where the customer wants to avoid heavy customization of the OBI applications & warehouse and wants to implement a solution with minimum customizations, the reasons might be varying for this but we do come across such scenarios and then we need to ensure that in spite of all the constraints we are giving the best solution to the client and also the solution should be something that keeps the customer happy. Here we will discuss how we can optimize the solution by utilizing the available Hyperion/Essbase and OBIA resources in order to get the best of both worlds. To start with let's have a look at the various technologies in picture here.

TECHNOLOGIES/TOOLS IN USE

Following is a brief description of the various tools and technologies being used in this solution.

1. Oracle BI Apps: - OBIA is a pre-built solution from Oracle which leverages OBIEE platform/ architecture to deliver horizontal and vertical business applications. OBIA is a - complete, prebuilt BI solution that delivers role-based intelligence for all the employees in an organization—from front line employees to senior management—that helps to ensure better decisions, actions, and business processes. The Apps can work with a range of data sources and applications that include Oracle E-Business Suite, PeopleSoft, Siebel, JD Edwards, and third party systems such as SAP. They are available for various business verticals & horizontals with prebuilt solutions specific to the vertical or horizontal.
2. Oracle Hyperion Essbase: - Oracle's Hyperion Performance Management applications are a suite of integrated applications that support a wide range of performance management processes and enable management excellence. It includes both strategic and financial processes. This suite drives profitable growth by delivering predictable results, improving transparency and compliance and increasing business alignment. The suite offers powerful, integrated reporting and analysis tools, including production reporting, graphical analysis and a MS Office interface built for use by end-users. Oracle's Essbase is the industry-leading OLAP (online analytical processing) server. It provides an environment for rapidly developing custom analytic and enterprise performance management applications. With a rapid application development environment managed by the business, Oracle Essbase enables the business to quickly model complex business scenarios. It is hot-pluggable across any data source.
3. DIM (Data Integration Manager): - It is the tool supported by Hyperion for data movement, it can be referred as Informatica with built in adaptors for Essbase. We can use DIM for data movement or metadata movement depending on the requirements. It can pull data from the Essbase source or populate Essbase targets. Also for Essbase as Target we can get data either from EBS or any other transactional data source say JDE/Siebel or we can also use OBIA warehouse depending on the data which is available in the warehouse and the kind of reports the user is looking for. If there are multiple Essbase applications then we can also use the Integration approach to ensure metadata movement from one cube to another and use it for Metadata Management Process. We can also use ODI (Oracle Data Integrator) another Integration product in Oracle stack.

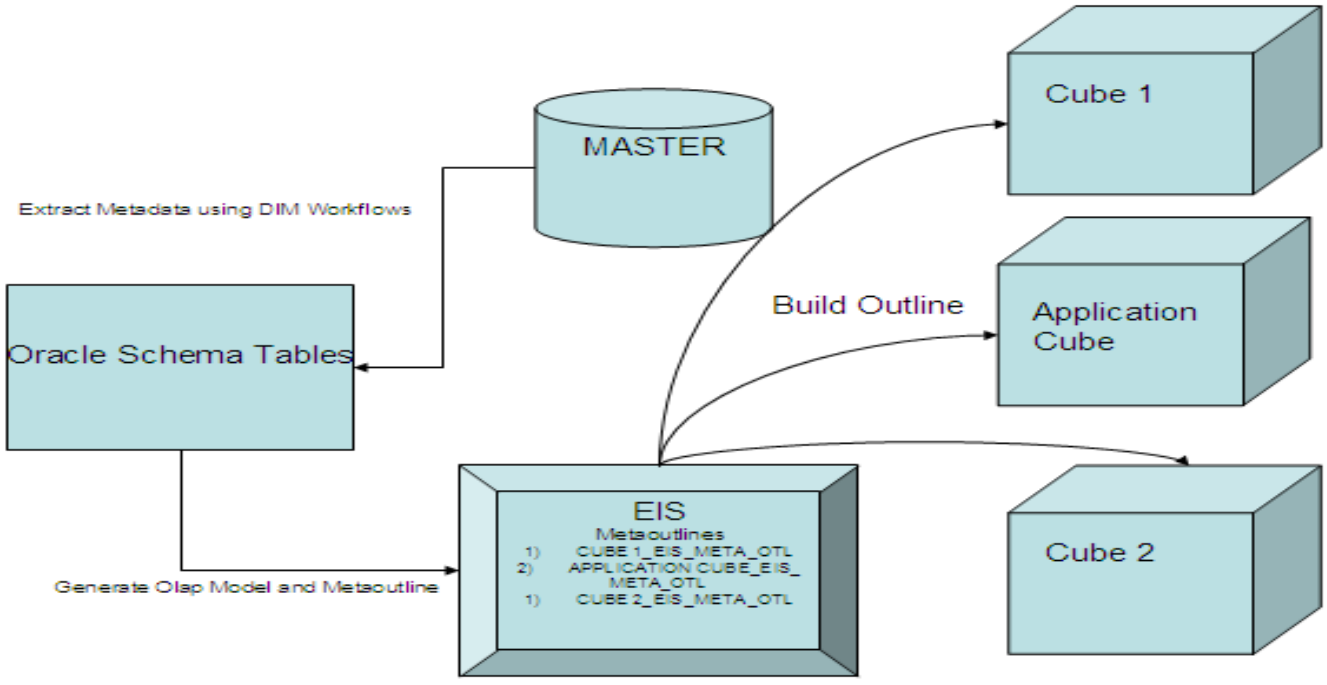
4. Essbase\Hyperion Analytics Integration Services (EIS): - It is an optional product in the Oracle Hyperion stack, it provides a metadata driven environment to establish a relation between data stored in Essbase databases and detailed data stored in relational databases. It allows users to drill down to relational database from aggregated data stored in Essbase. It will help to create and derive the database and it's outline and also stores metadata about the OLAP model.
5. Analytical/Essbase Administration Services (EAS/AAS): - It is the database and system administrator's interface to Essbase. It allows users to design, develop, maintain and manage multiple Essbase servers, applications and databases.
6. Essbase Spreadsheet Add-In for Excel – The Add-In integrates Essbase with Microsoft Excel. It adds Essbase menu to Excel which provides commands such as Connect, Retrieve, Pivot etc. in Excel. Users can access and analyze data by simple mouse clicks by using this Add-In.
7. Oracle: - Oracle is the integrated database here, but the configuration is not database dependent and we can use any of the databases apart from Oracle also.
8. Source/Targets: - We can have data/metadata coming from any of the Hyperion applications like HFM, HSF etc and also the OLTP application can be anything like JDE, Siebel CRM, PeopleSoft or even SAP, but the source transaction system in picture here is EBS.

Solution Approach

We will discuss two of the main scenarios here, *one will be the metadata management process in a multi cube setup* and another will be *the usage of Drill Through Reporting on data warehouse from Essbase* using the above said tools and technologies.

Metadata Management Process

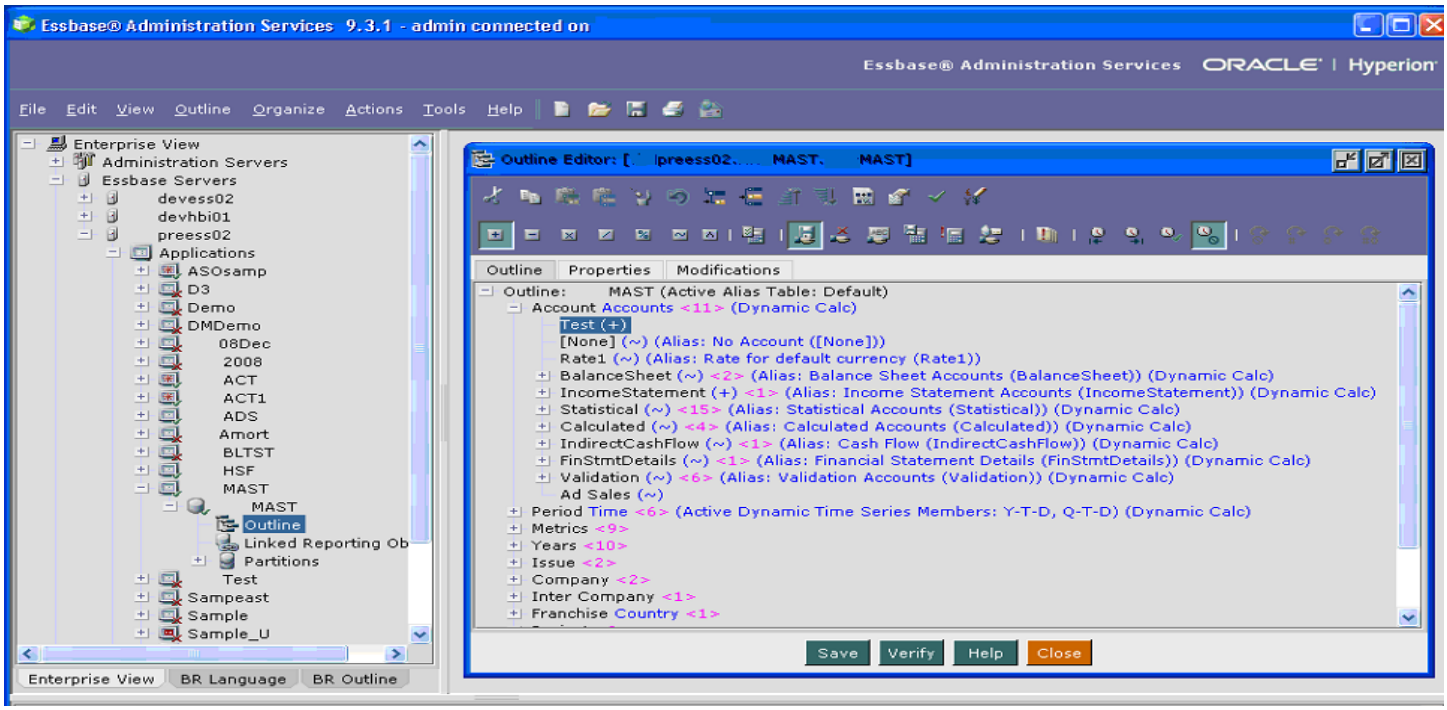
One of the constraints we normally face in a multi cube environment is the metadata management, as there might be multiple cubes and if there are frequent metadata changes then it becomes bit difficult to keep all the cubes in sync unless we don't have a proper metadata management process in place to ensure all the cubes are up to date and current with the latest details. Here we will discuss one of the solution approaches which can be used for the metadata management process. The metadata source can be any Hyperion application depending on the environment be it HFM, HSF etc. We can use the approach of having a single master cube which can be created with the data from the applications in use and then going forward all the future changes should be made in this master cube and then made to flow to the other respective cubes in use. The following diagram will help with the better understanding of the process before we go into the details.



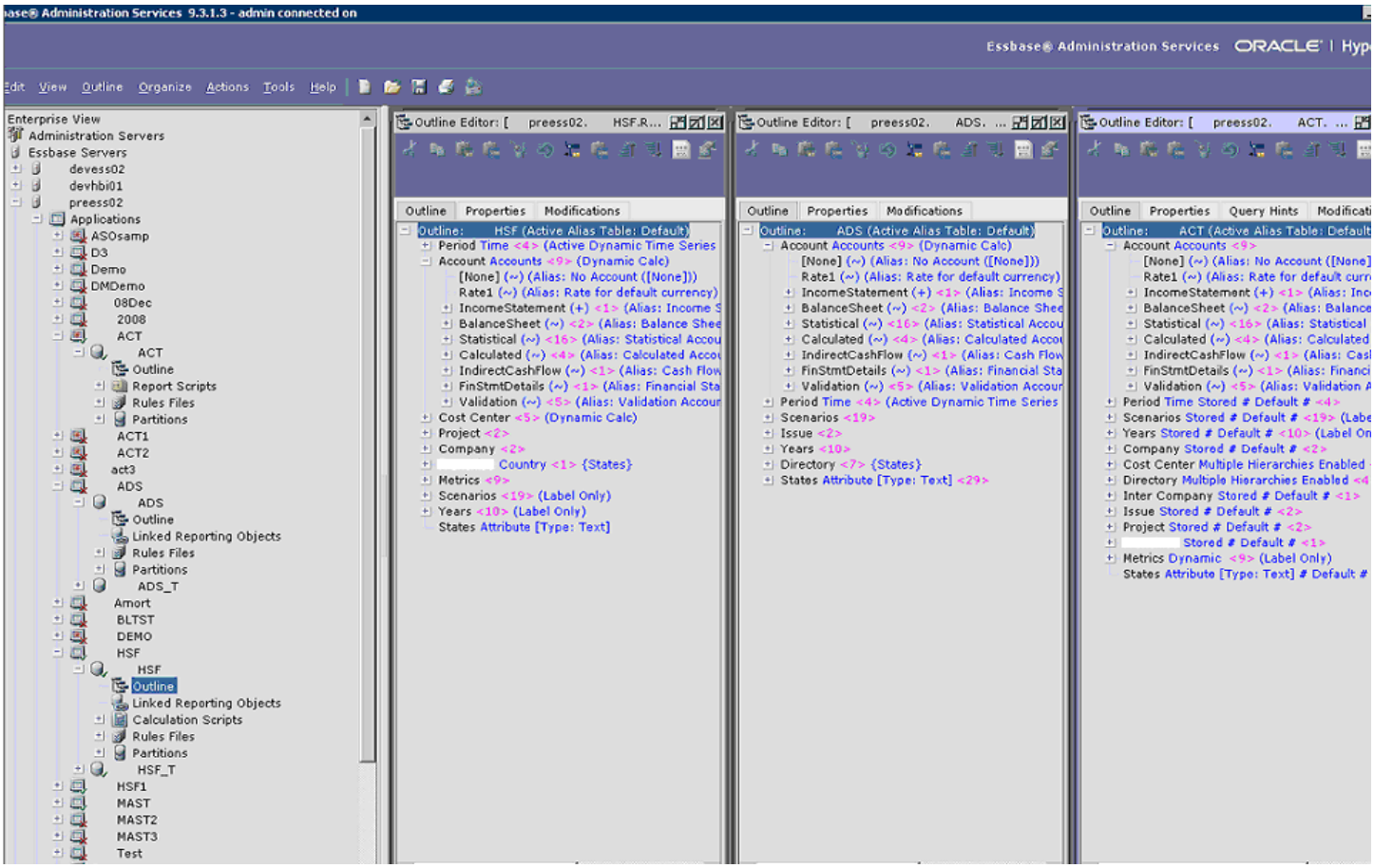
In the above shown setup there is a master cube which has the metadata that has been obtained from the source application using a one time process, the source master cube then populates the metadata into the Oracle OLAP schema using DIM or say ODI and then we have EIS which is being used to flow the data from the Oracle schema tables which are populated with the metadata in to the required cubes.

Next we will discuss the various requirements and steps to implement this solution.

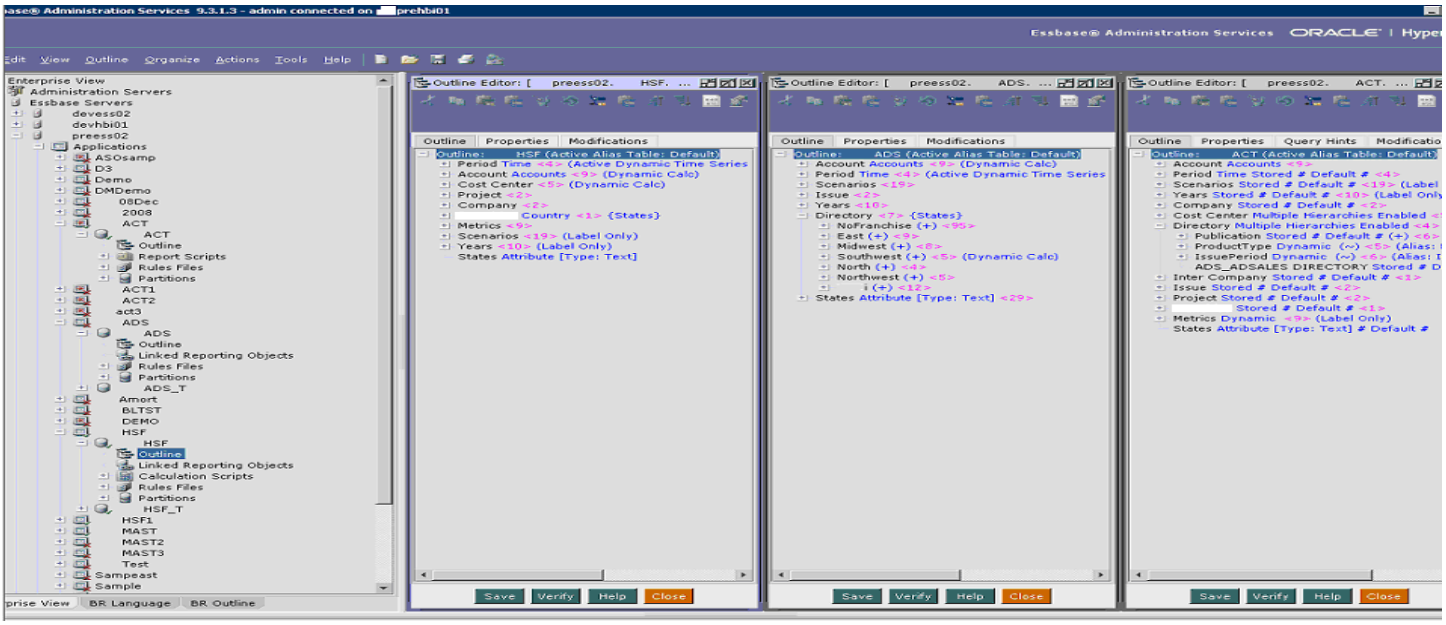
1. One of the requirements here is all the cubes should have similar dimensions with similar members.
2. The second requirement is that the master or the source cube should have all the members of all the cubes. All the metadata should be first updated in the master cube and then flown into the various other cubes. The screen shot below shows the Outline of the Master Cube.



3. The screenshot below shows the members of the three cubes which we are using.

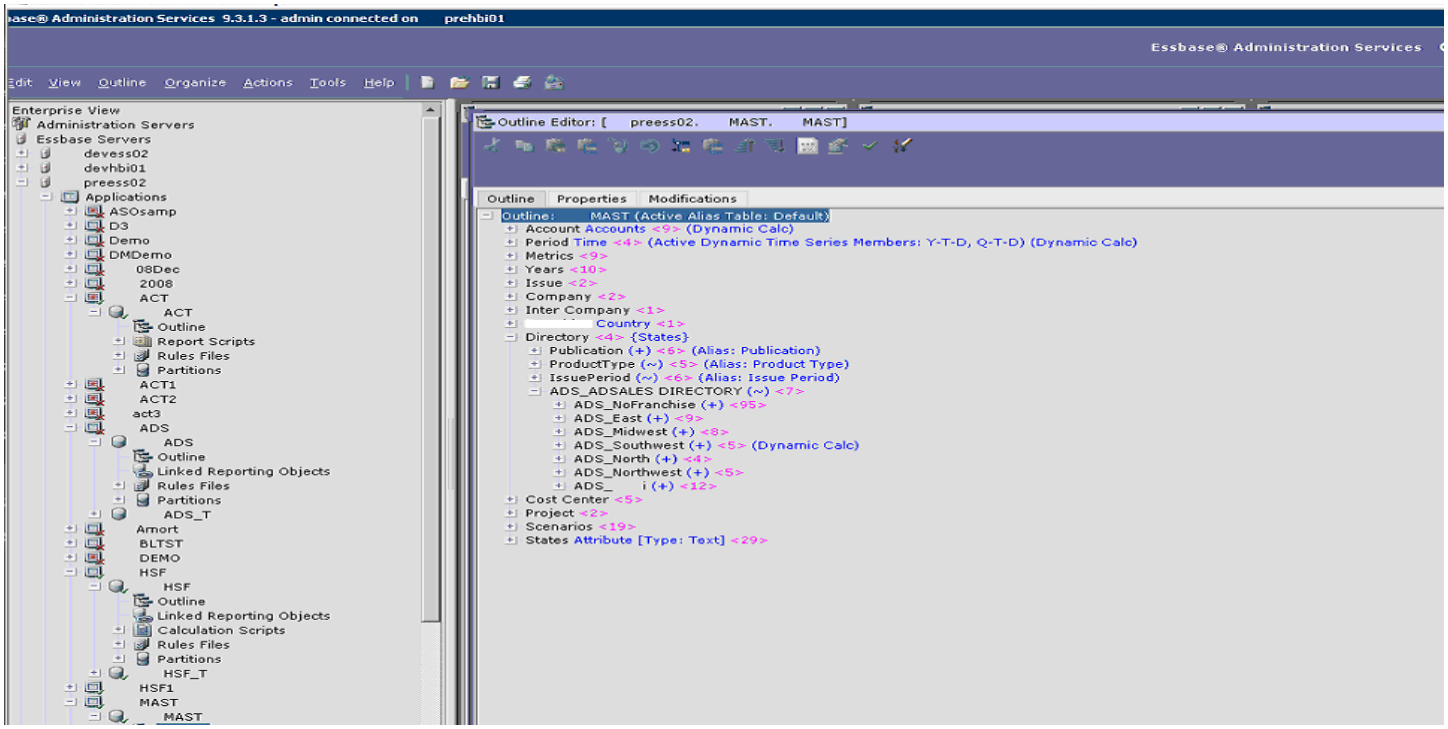


4. Here above we are seeing the Outline of the three cubes which are being used in this solution and we need to have access to EAS to check & update the outlines for the cubes as per the requirement.
5. We might run into issues where a few dimensions or the members might be different and such issues need to be handled separately, if the differences in dimensions are huge then it's as good as creating 3 different cubes and the approach of using single cube to update all the cubes will not serve the purpose. The easiest way to solve this issue is to filter them out while populating the respective dimension tables.
6. The approach of having a single Master Cube might not be feasible in case the multiple cubes in picture are all different in themselves; we need to have cubes with similar data to make this approach work. If there are several cubes and there are similarities between them then instead of a single master cube we can have 2 set of cubes depending on the number of cubes to be maintained.
7. If there is a member that needs to be there in two cubes but with different values then we can have 2 different members in the Master cube as the Outline will expect unique members. These members can then be filtered in DIM while populating the OLAP tables, similarly in the OLAP table we will have to create two different tables for each of the dimensions of the different cubes. The screenshot below demonstrates this with the example of Directory dimension which is different in two of the cubes.



8. We need to have a Star Schema with all the facts and dimensions (Oracle Schema tables) in the above example defined for this approach as EIS mandates a relational Star schema.
9. We will also need an Integration Tool say DIM or ODI to flow the metadata from main cube to the relational tables, so that EIS can access it to update the various application cubes.
10. Next we can discuss the various steps to implement the solution.
11. We will start with the various steps to create the Model & Meta outline for this solution.

A. The first step here will be to have the Master Cube ready with the metadata, which is going to be the main source for all the cubes. Populating the Master Cube will depend on the existing applications in use and it might be a one time process to populate the Master Cube from the applications like HFM etc. We need to access EAS to be able to look into the Outline of the cubes.



B. Outline – Outline describes the structure of the database and has the details of the various hierarchies, dimensions, measures and key performance indicators

C. We need to ensure that there are unique members in the outline, as we might come across similar or duplicate members and we will need to name them accordingly. In the similar fashion even OLAP tables will have need to have two different tables for each of the dimensions of different cube.

D. Next we need to have the corresponding Star Schema designed in the relational schema/database. This schema will be used as a source by the EIS. We will need to flow the metadata from the Master cube to the relational schema. And that is where we will use DIM (Data Integration Manager).

E. For this integration we will need infrastructure for DIM (Data Integration Manager – Informatica) which comes with Hyperion or we can also use ODI or Informatica. DIM as such can be referred as Informatica with Essbase adaptors and it comes bundled with Hyperion.

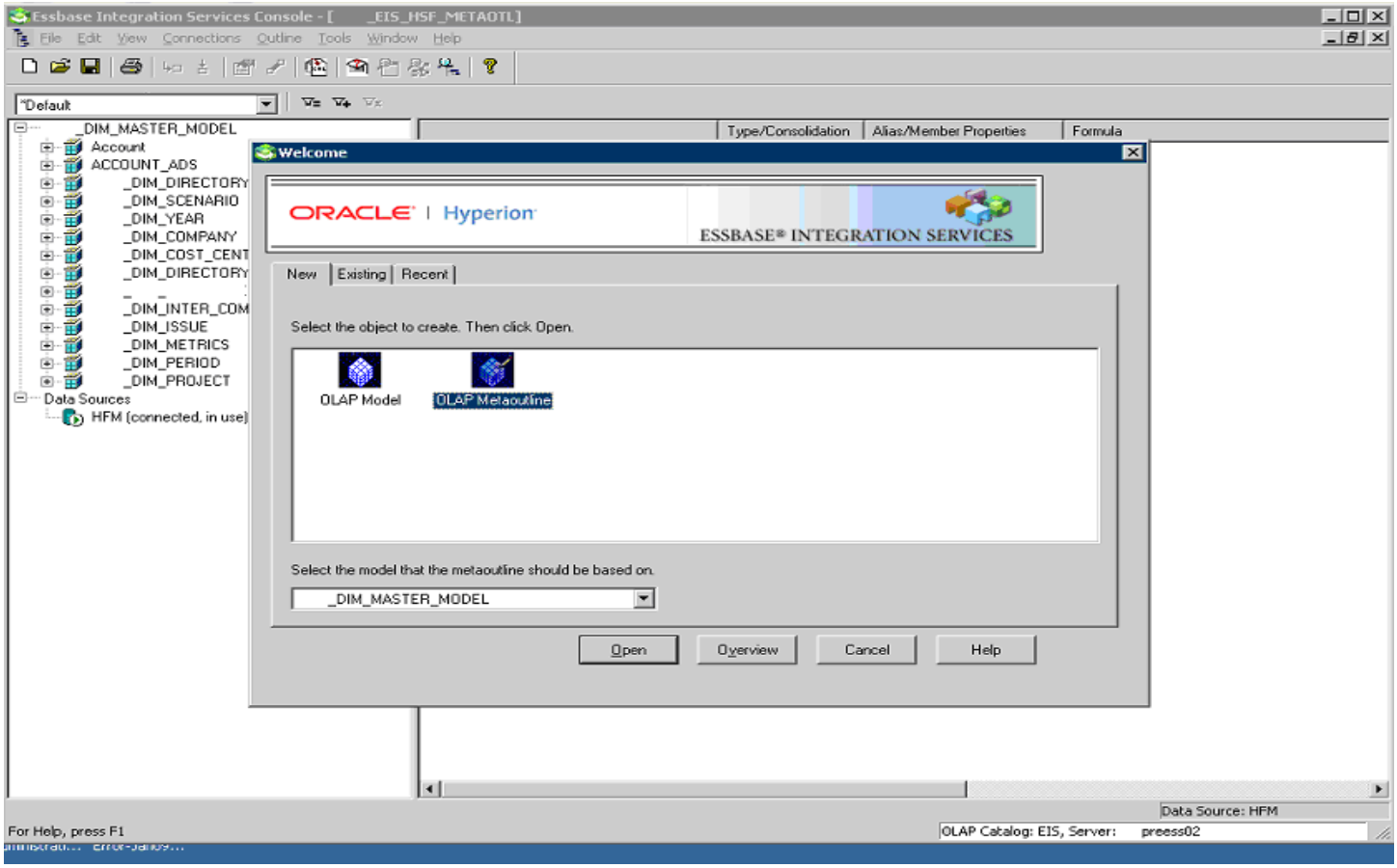
F. If we are using DIM then it will involve creation of Source/Targets, Mappings, Mapplets, Sessions and Workflows. The process will be used to extract metadata from the master cube and load it into the OLAP tables. And once the meta-data has been loaded into the Oracle tables we can use EIS to load the same into the reporting cubes.

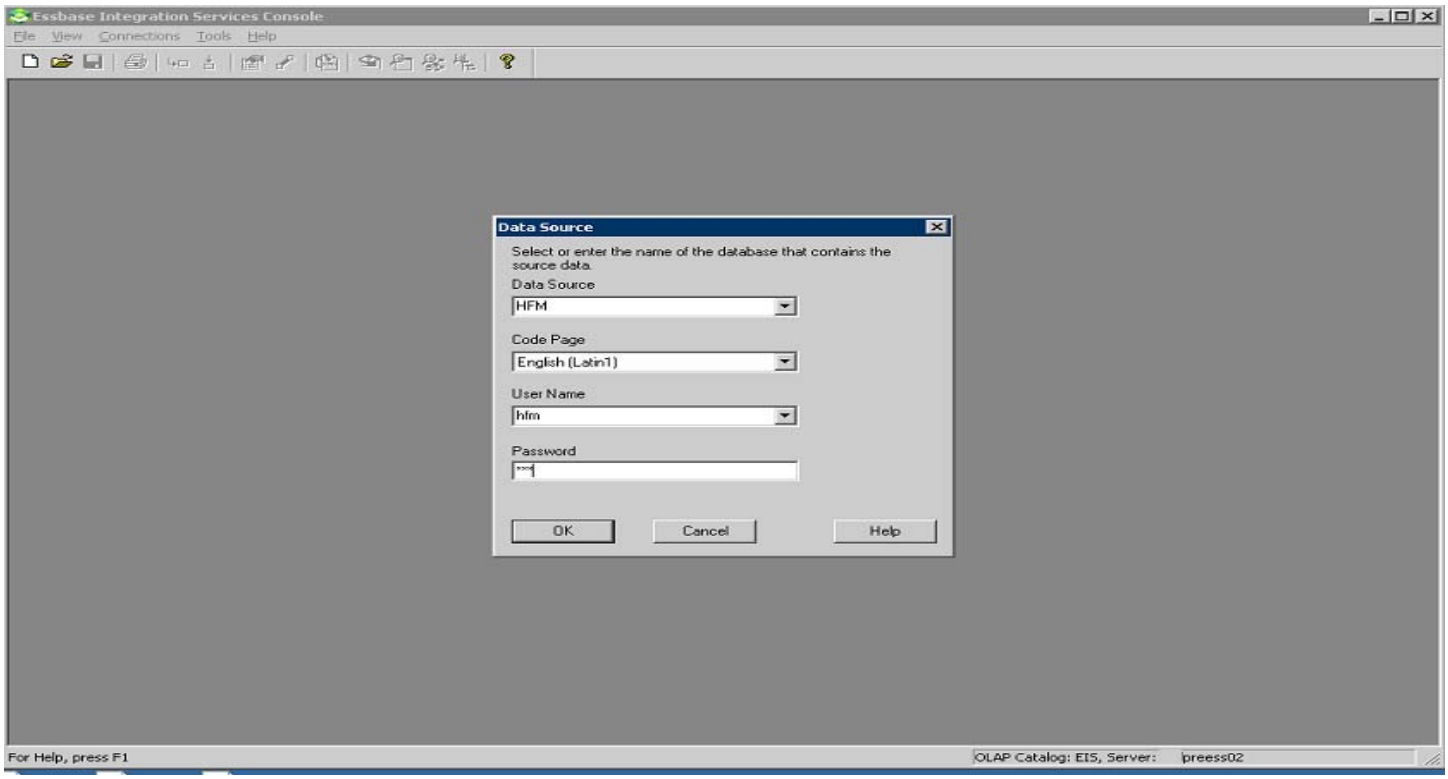
G. Here in this case we will have Essbase Cube as the Source and the target is going to be the relational database the OLAP model which we created, so the Integration tool in use should be able to support Hyperion Essbase.

H. Next in order to create a Master model we need to be logged in to Essbase Integration Services. In our solution approach we have one master cube and three reporting cubes, so we will need to create one master model and three meta-outlines.

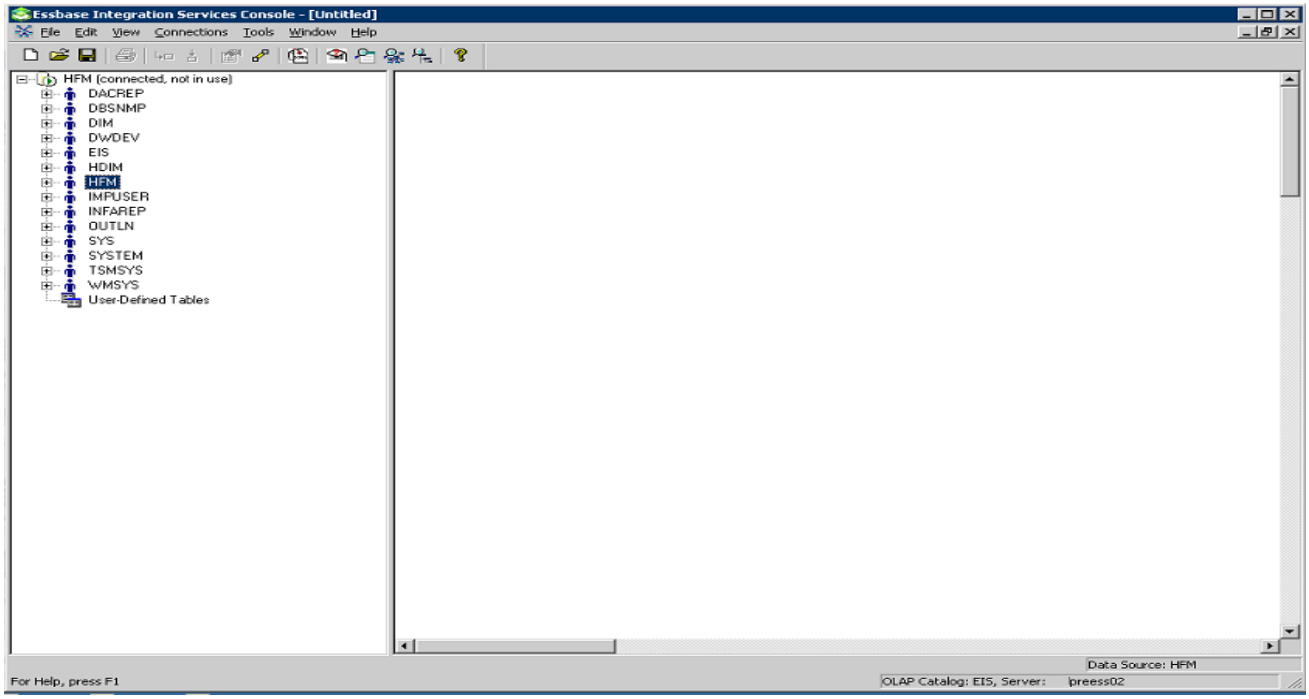
I. Once logged into EIS the user can either select an existing model or create a new model as per the requirement.

J. On selecting New we will get an option to create new OLAPModel or OLAP Metaoutline. We will select OLAP Model here. And next we need to select a data source where OLAP data model exists in the relational database. After selecting the data source it will open a screen which will have all the listed schemas/users and we need to expand the schema we need to use.

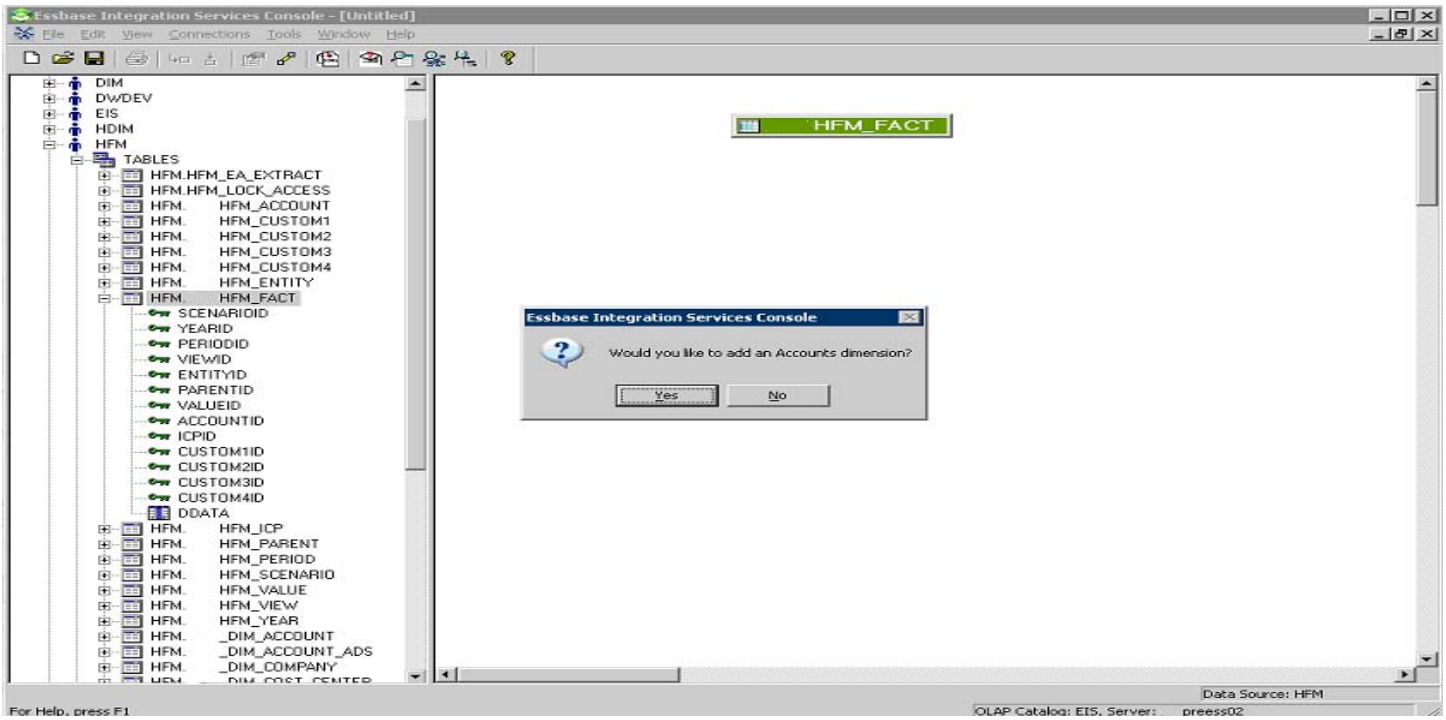




K. The expanded schema will have all the fact and dimensions under it.

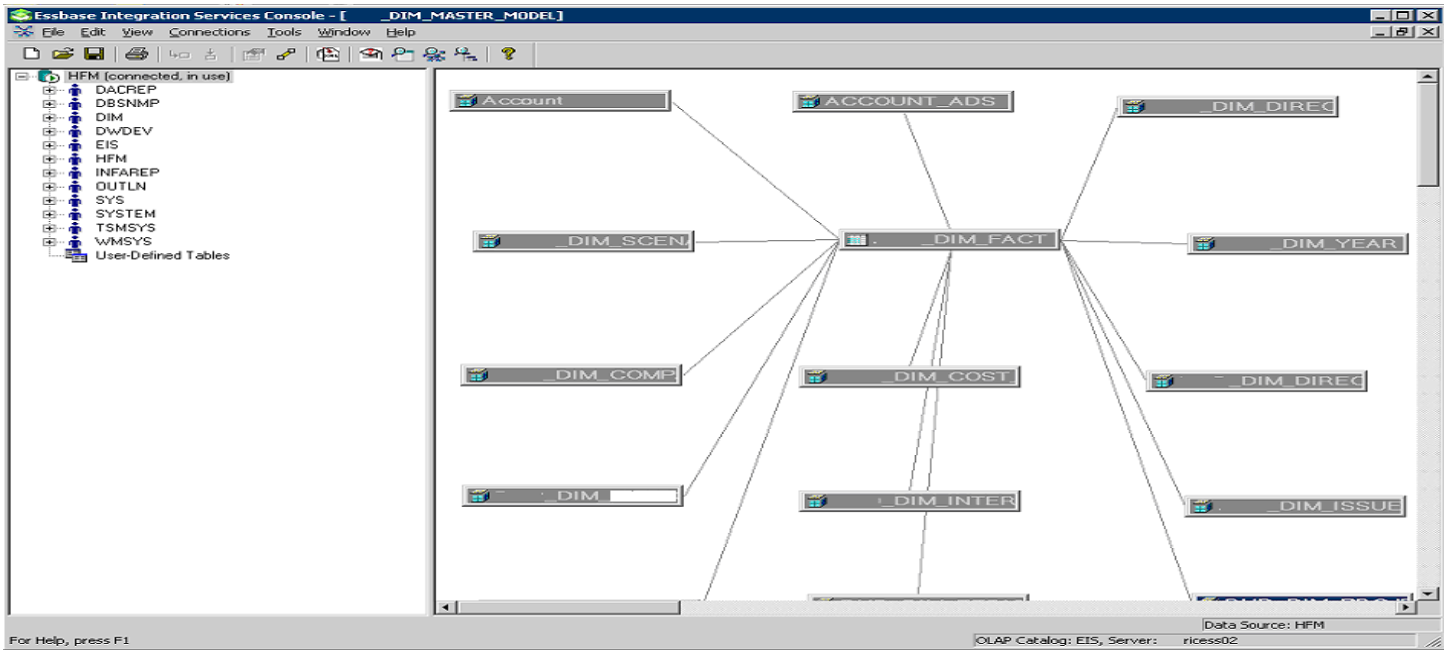


L. In order to create the model we need to drag the respective tables listed under the schema on the left to the right side, we can start by extracting the fact first and then pulling in all the dimension tables one by one. We will also need to create the joins between the fact and dimensions by selecting the required columns.

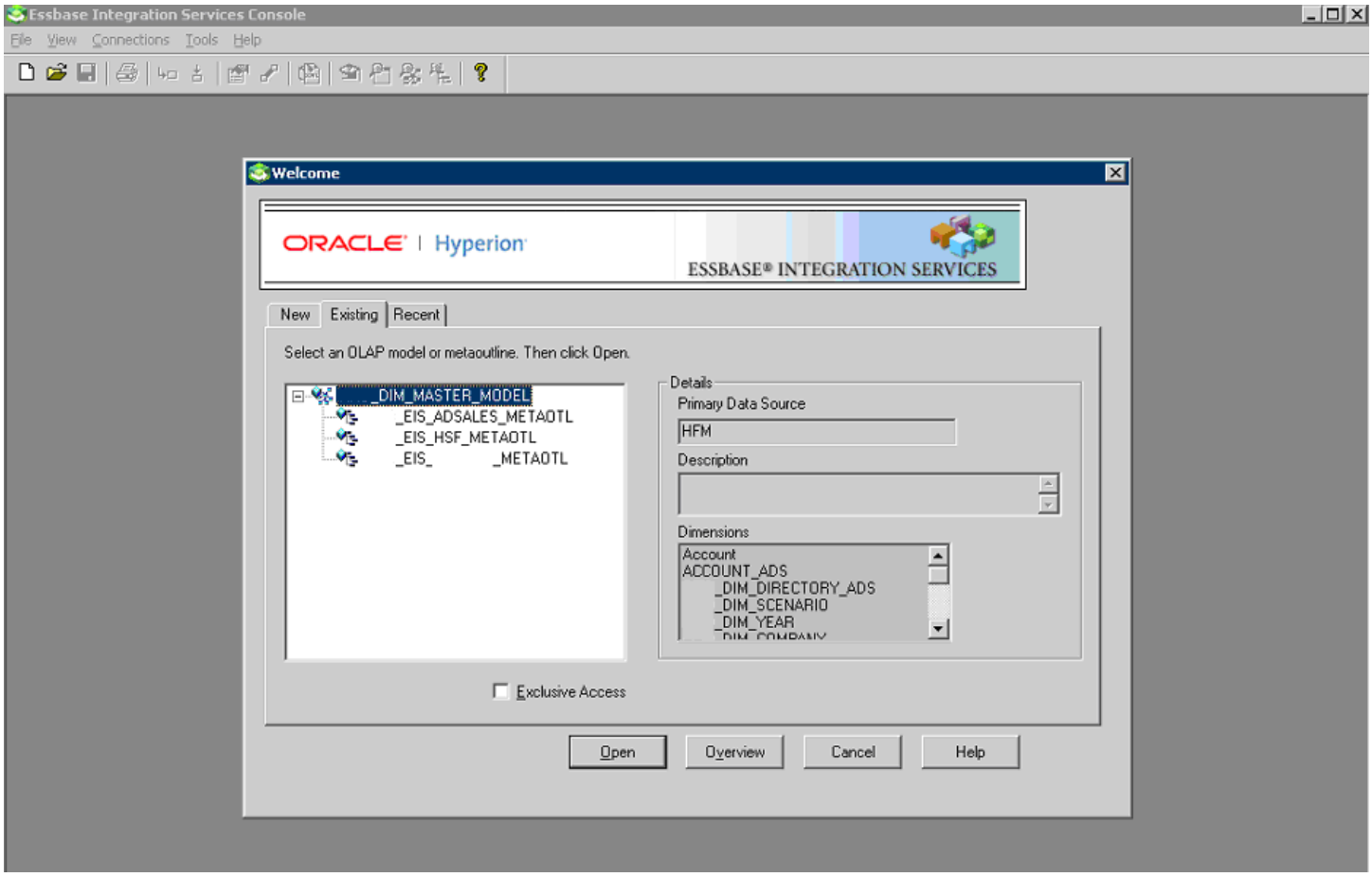


M. We will need to pull down all the dimensions from the left side to complete the model with the respective joins.

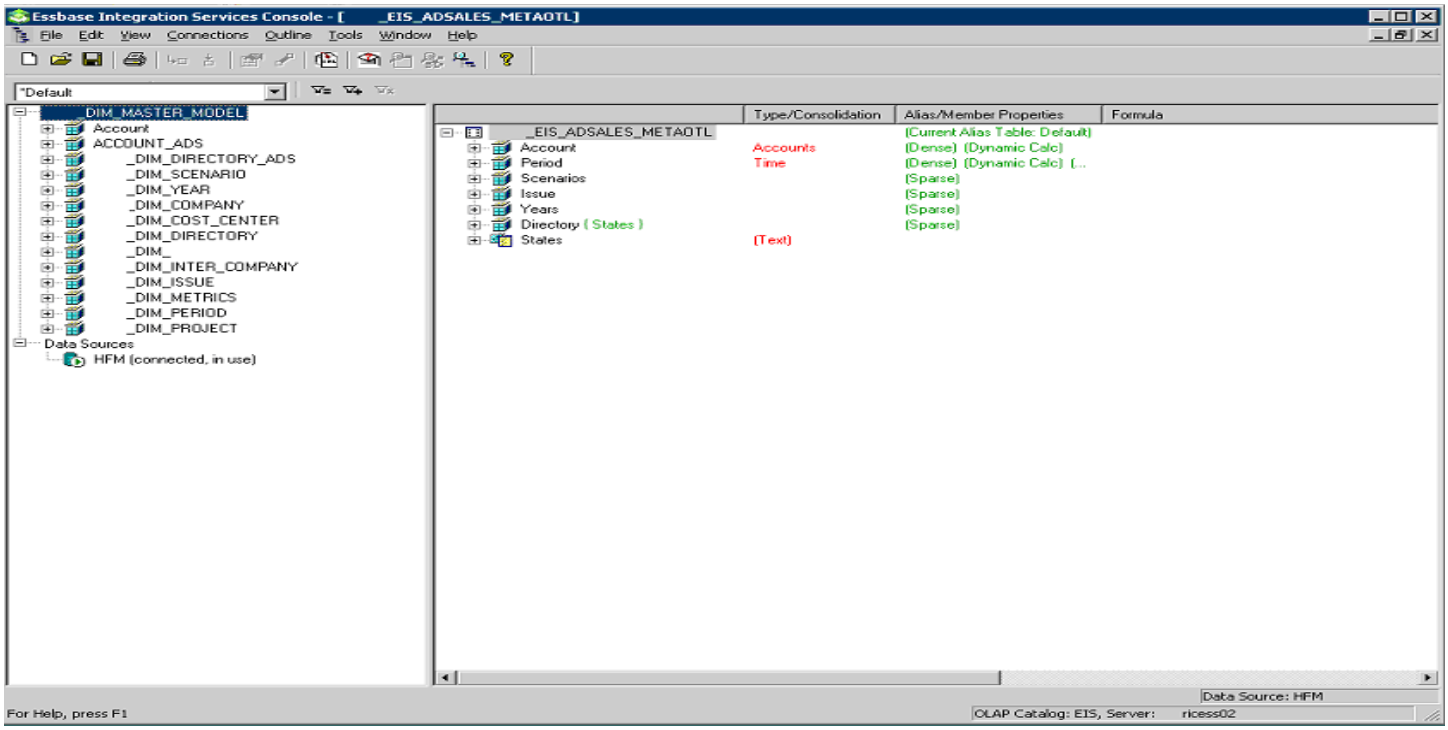
N. After adding all the dimensions and the fact our Master Model will be ready, the screen shot below will show us how a Master model might look like.



O. Next we need to create the Meta-Outlines and depending on the dimensions in the corresponding cube we need to use the respective tables. We need to select New OLAP Meta-Outline and select the recently created Model, the meta outlines will be based on the Master Model.



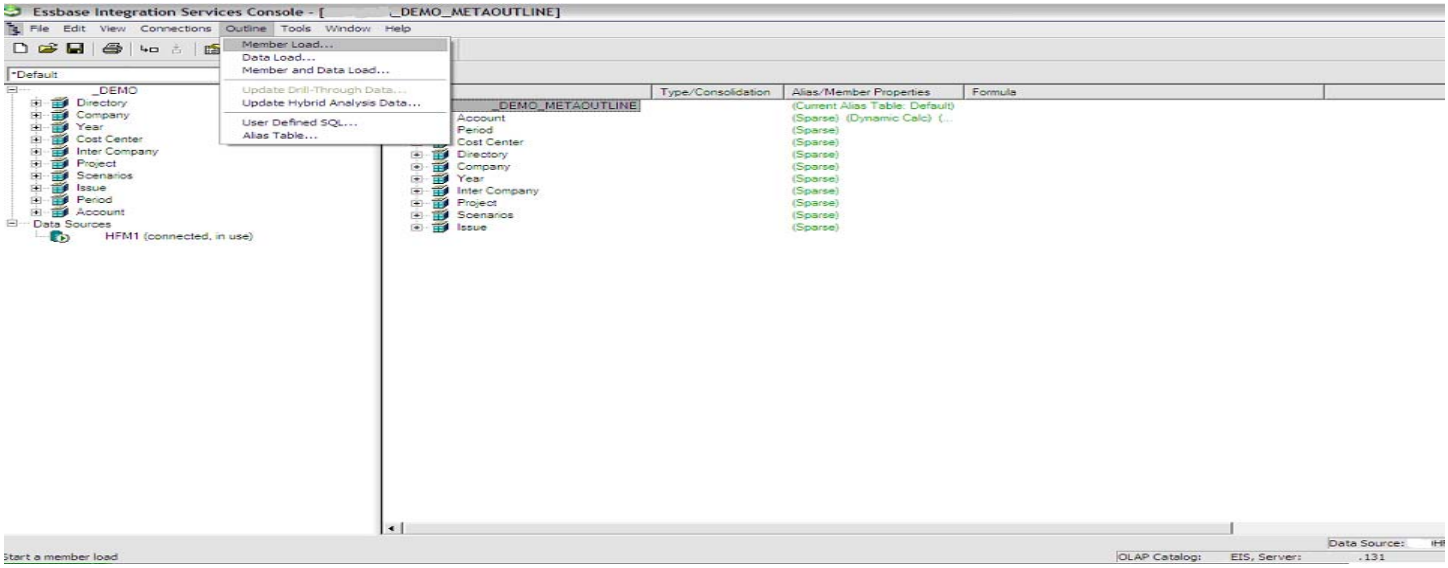
P. The meta-outline can be created by dragging the members from the OLAP master model into the right side. The screen shot below will show a meta-outline.



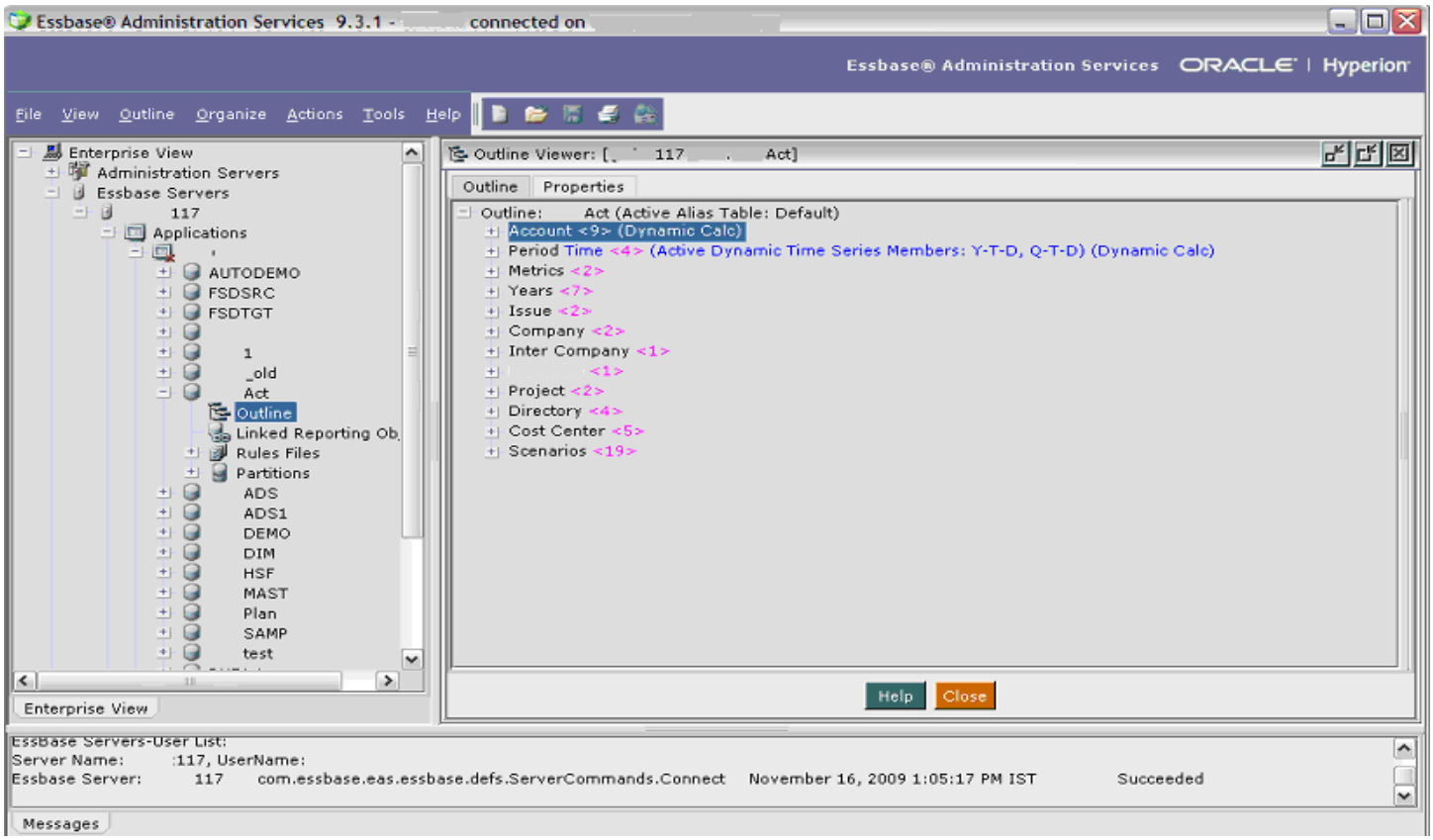
Q. Once the meta-outlines have been created the next step is to load meta-data into the cubes. Now we might run into ASO or BSO cubes in this process.

R. Now to initiate the load process we need to login to EIS and select Existing tab out of the three tabs. In our case we will see three meta-outlines (one for each cube) under the OLAP model. We will need to select the respective meta outline and click on Open. Next we need to enter the data source user id and password.

S. Once in the meta-outline we need to select Outline→ Member Load from the main menu.

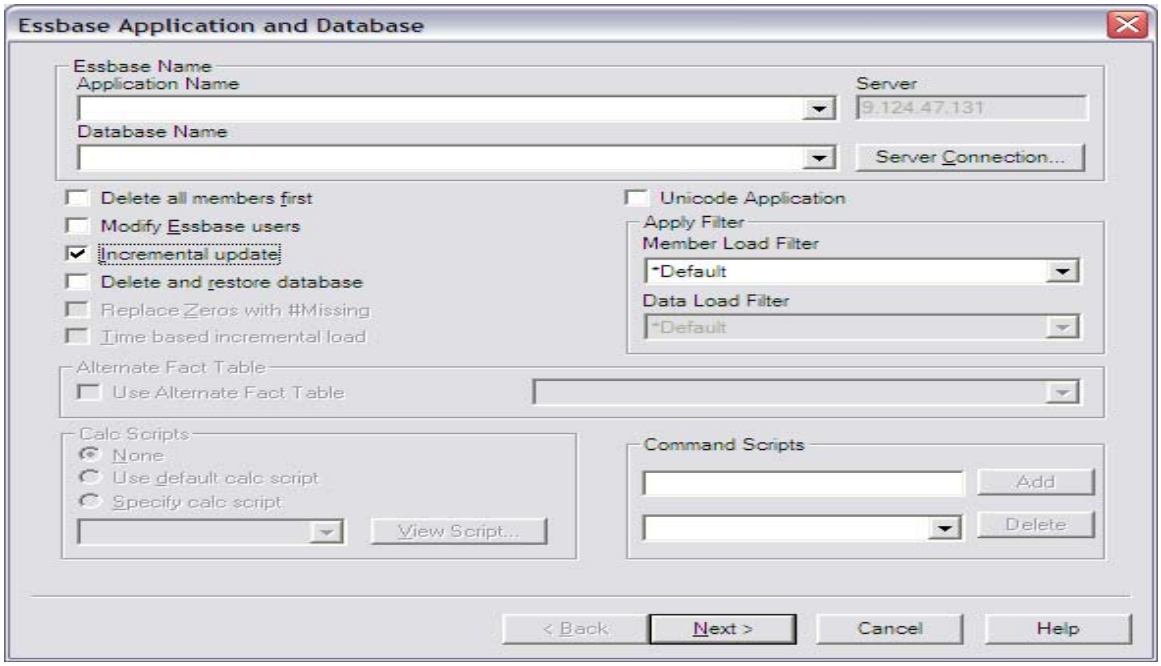


T. Next we need to go to the Essbase Administration Service and open the desired Outline in the application and database on which member load needs to happen.



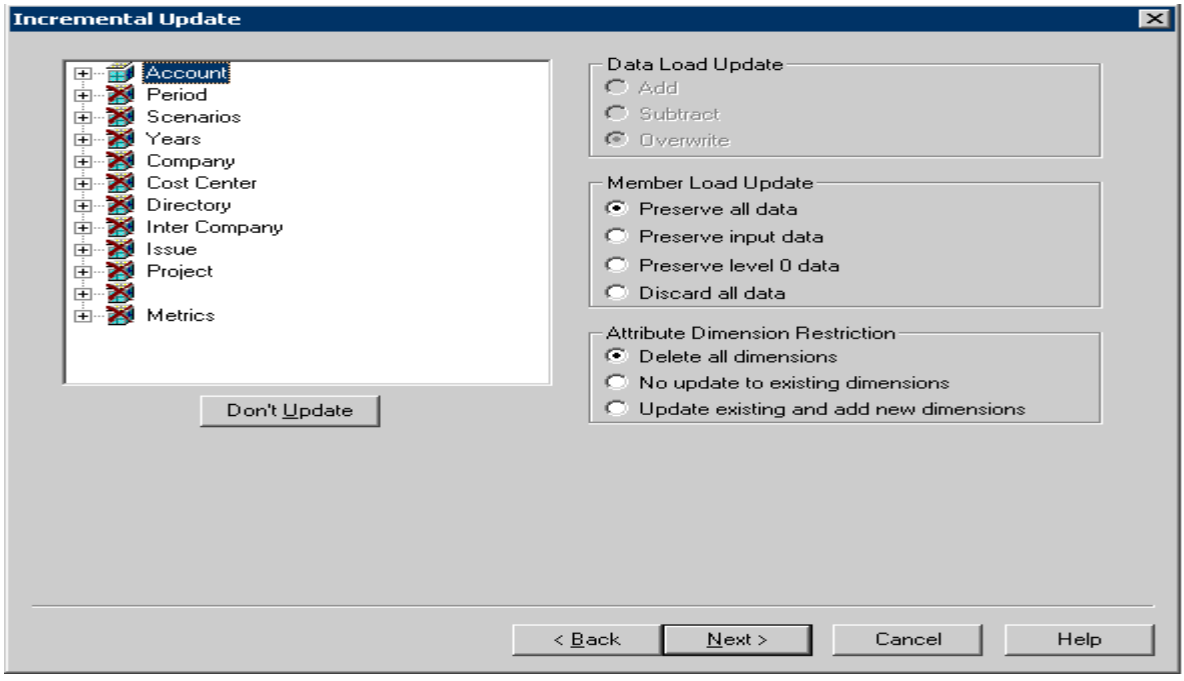
U. In case there are shared members in a dimension and we are loading data into ASO cube then we need to delete the dimension before the Member load. If we don't delete the dimension then we will have double the number of shared numbers. We don't need to delete all the dimensions of the application but only those having shared members.

V. Also we should not be doing Delete All Members in ASO cubes with shared members as apart from deleting the members from the Outline it might also lead to corruption of the Outline as the load might not be able to create the Outline and the outline will get invalidated. This is something based out of experience and may or may not happen in different versions. A simple solution for this would be to go for Incremental update while loading ASO cubes.

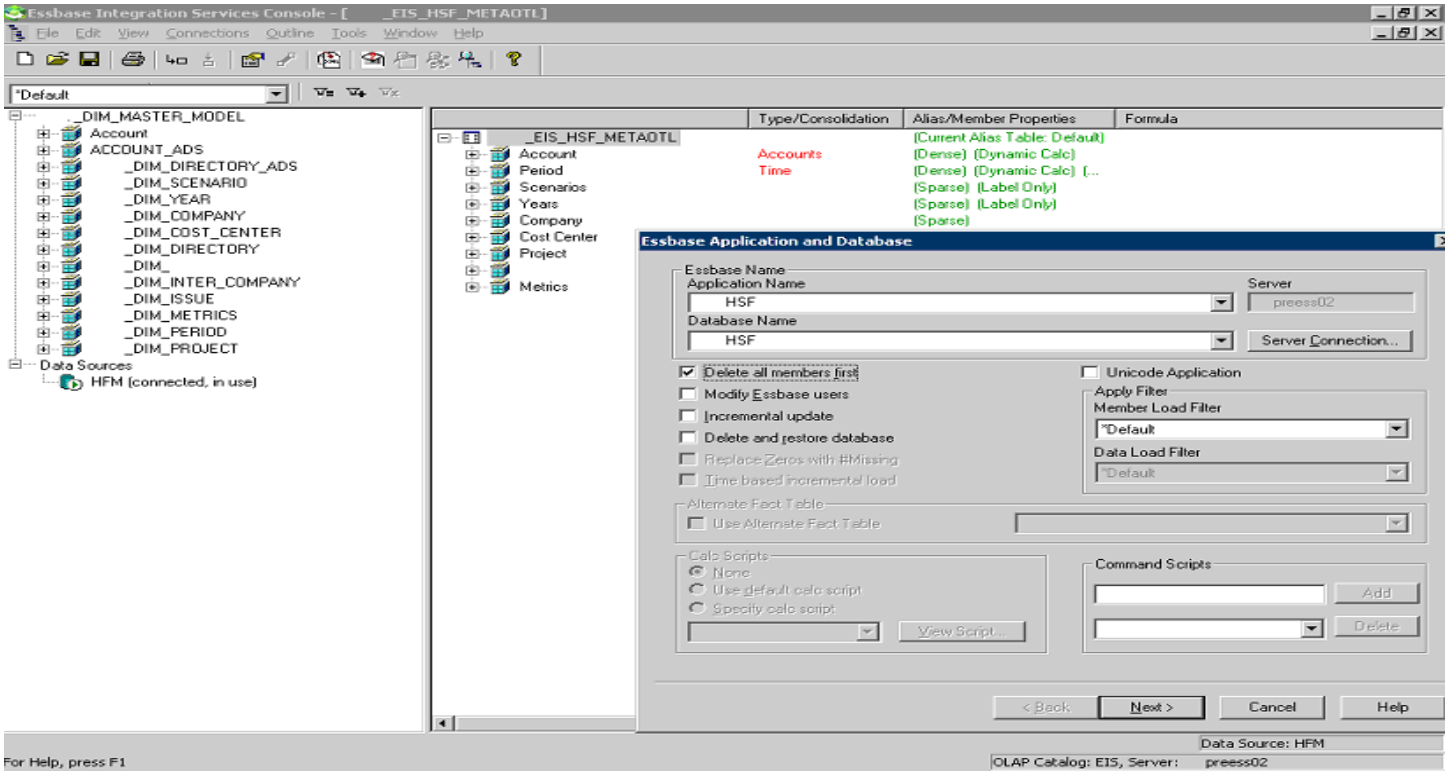


W. If we need to update all the dimensions in an application then we can delete the respective dimensions with shared members and run an incremental update on all the dimensions.

X. Also we might need a selective update on the dimensions in the application so in that case we will need to select "Don't Update" for the other dimensions which don't need to be updated.



Y. For a BSO cube or ASO cube without the shared members the load is pretty straight forward and we need to select delete all members in EIS, this is required to capture parent changes in the particular members.



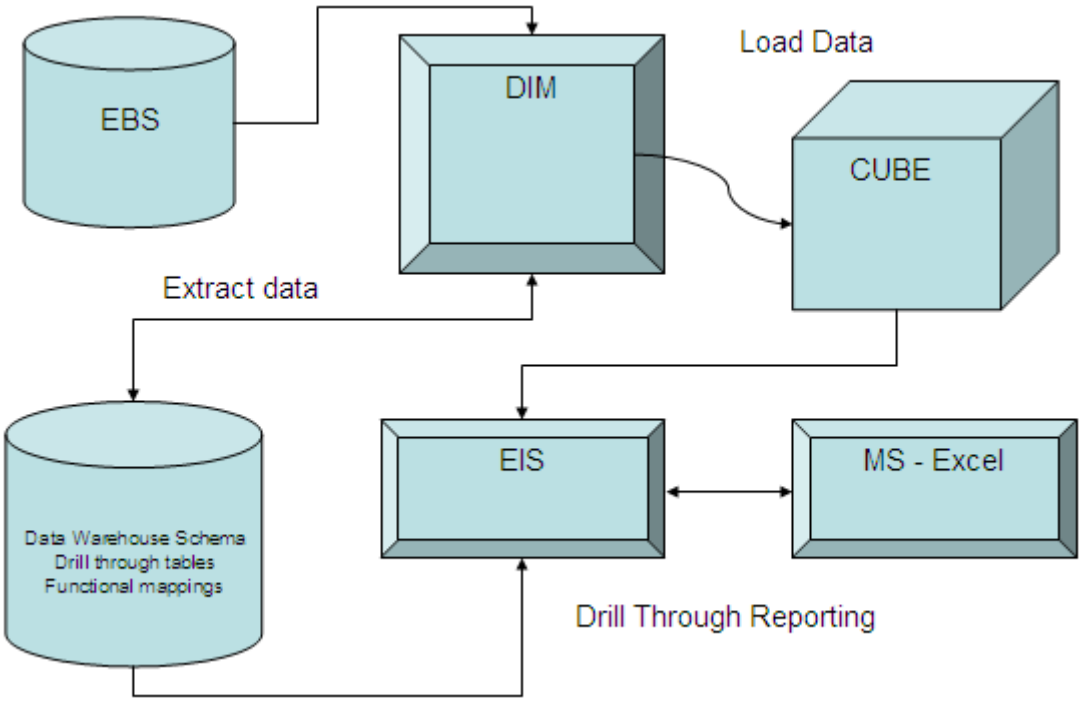
Z. Once the load is over there will be message in the EIS console saying "Successfully loaded <Server> :< Application>", this message will ensure the successful loading of meta-data into the required application.

AA. This very much finishes the metadata management process using a combination of technologies from the Oracle stack.

DRILL THROUGH REPORTING

In the current business scenario we generally come across customers having setup of Hyperion Essbase and ERP applications like JDEdwards, EBS or CRM and also planning to implement OBIEE or even having both say Hyperion Essbase and OBIEE apart from the ERP implementations and they will prefer leveraging the existing environment with minimum investments and maximum returns. And in such scenarios we will need to provide a solution that uses the best of both worlds and involves minimum customization efforts and investment. Here we will discuss a case study involving Hyperion / Essbase and Oracle Business Intelligence Applications (OBIA). We will look at leveraging the best of both the platforms to take advantage of Multidimensional capabilities of Essbase and pre-built data warehousing and BI capabilities of OBIA. We will look at the scenario where the customer manages account hierarchy and aggregated data in Essbase and the granular level data or detailed transactions in the Oracle Data Warehouse (OBAW). This is an ideal setting for drill-thru reporting. Often companies may want to maintain their complex account and organizational hierarchies in Essbase rather than in ERP systems like E-Business Suite.

This solution can be used when we have very little implementation time with requirement of minimum customization of the OBI applications and the warehouse. Here we will discuss how we can optimize the solution by utilizing the available Hyperion/Essbase and OBIA resources in order to get the best of both worlds. For drill through reporting we need to ensure there is relative data available in the OBIA warehouse as the aggregate sum available on Essbase should match with the transactional records. The following diagram will help with the better understanding of the drill through implementation process before we go into the details.



As per the diagram above we have data available at both the Essbase application cube as well as in the data warehouse tables. The data source for the Essbase application cube and data warehouse tables is EBS. For the cube we also have other disparate sources as an additional data source, in this case it is Hyperion applications which store the hierarchy information. The cube is used to store only the aggregated data and hierarchy information where as the detailed level data is available in the data warehouse tables. So here we have a requirement to allow users to drill down to the detailed level records based on the selection from the aggregated data in the Cubes.

Next we will discuss the various requirements and steps to implement this solution.

1. One of the very first requirements here will be to have relative data available in the OBIA warehouse so that the aggregate sum available in Essbase should match with the warehouse records.
2. The selection of ETL tool is optional and apart from DIM (Data Integration Manager) we can also use ODI available in the Oracle stack of products.
3. DIM will be used to extract data from the source, transform it as per the requirements and then finally load it into the targets, which will constitute of Oracle tables or Essbase Cubes depending on the stage of process.
4. EIS will be used to define the queries required for Drill-Through Reporting. The main business logic will go in here to define the query to drill down to the detail level records.
5. MS-Excel Add-In for Essbase will be required to drill down from aggregated Essbase records to detailed level data warehouse records.
6. Next we will discuss the various steps to develop this solution.

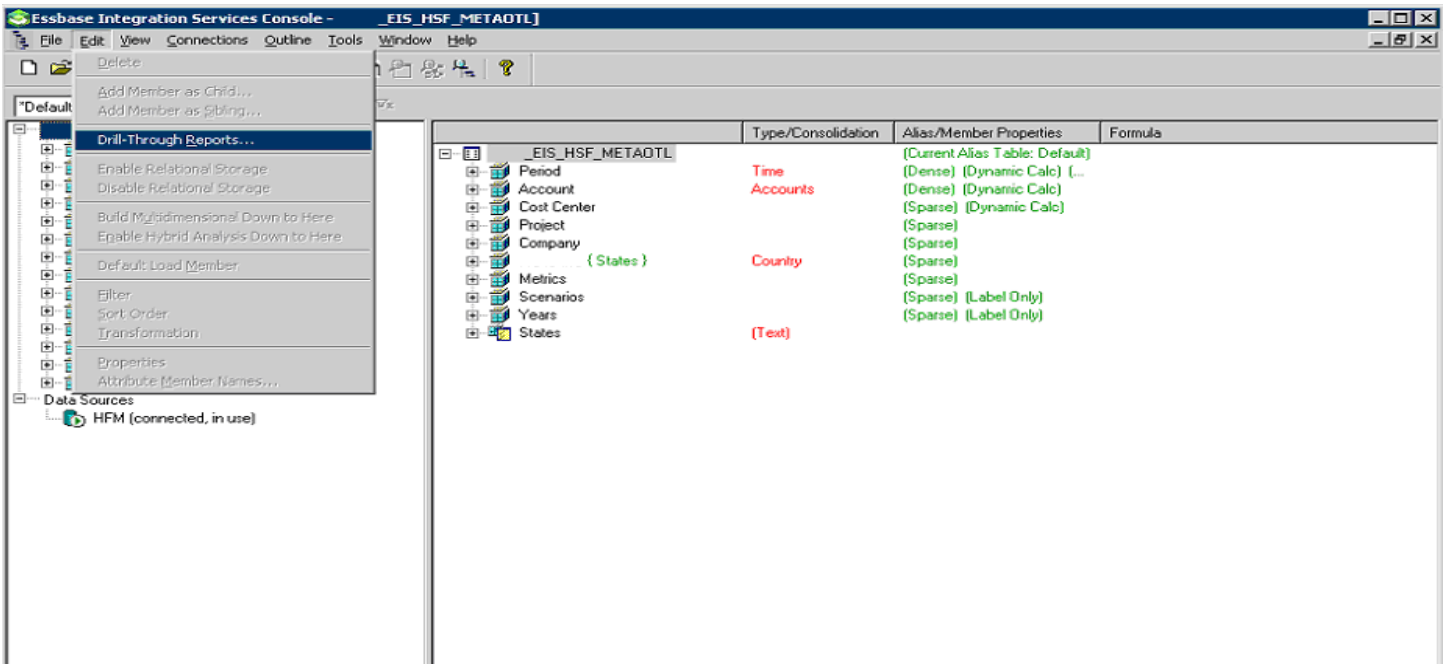
A. The first step as such will be the creation of the various mappings and workflows in the DIM to extract, transform and load data into the data warehouse tables and Essbase cubes. But since that being generic I will not go into the details of the same.

B. The next step will be the creation of the drill through structure in EIS.

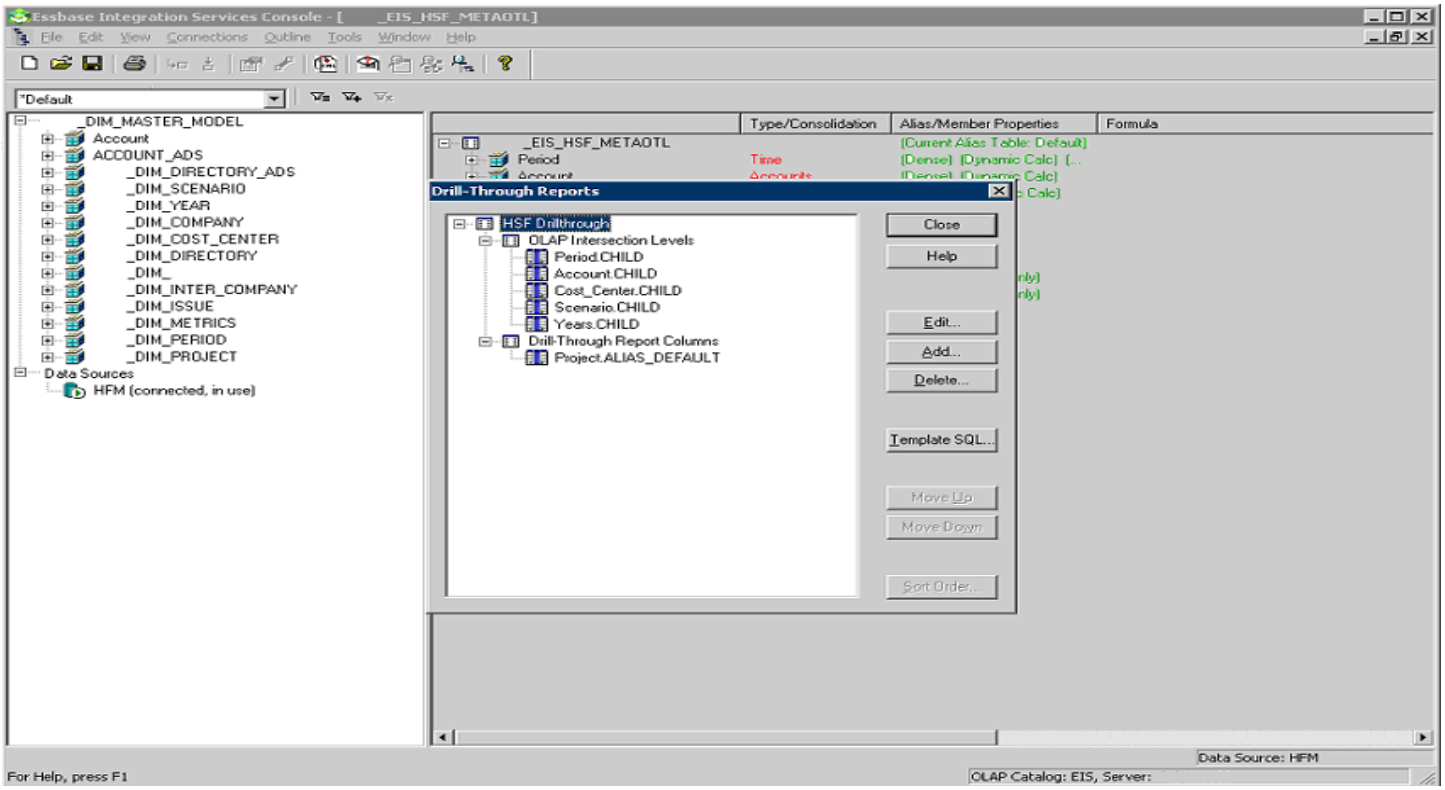
C. For this we need to login to EIS and select the OLAP Model and the Meta-outline. The OLAP Model and Meta Outline is something which we have already created during the Metadata Management Process, so we can directly use it else we will need to create it if we are using this solution independently.

D. Also we need to use one of the application cubes which will be having aggregate data and is relative to the relational data.

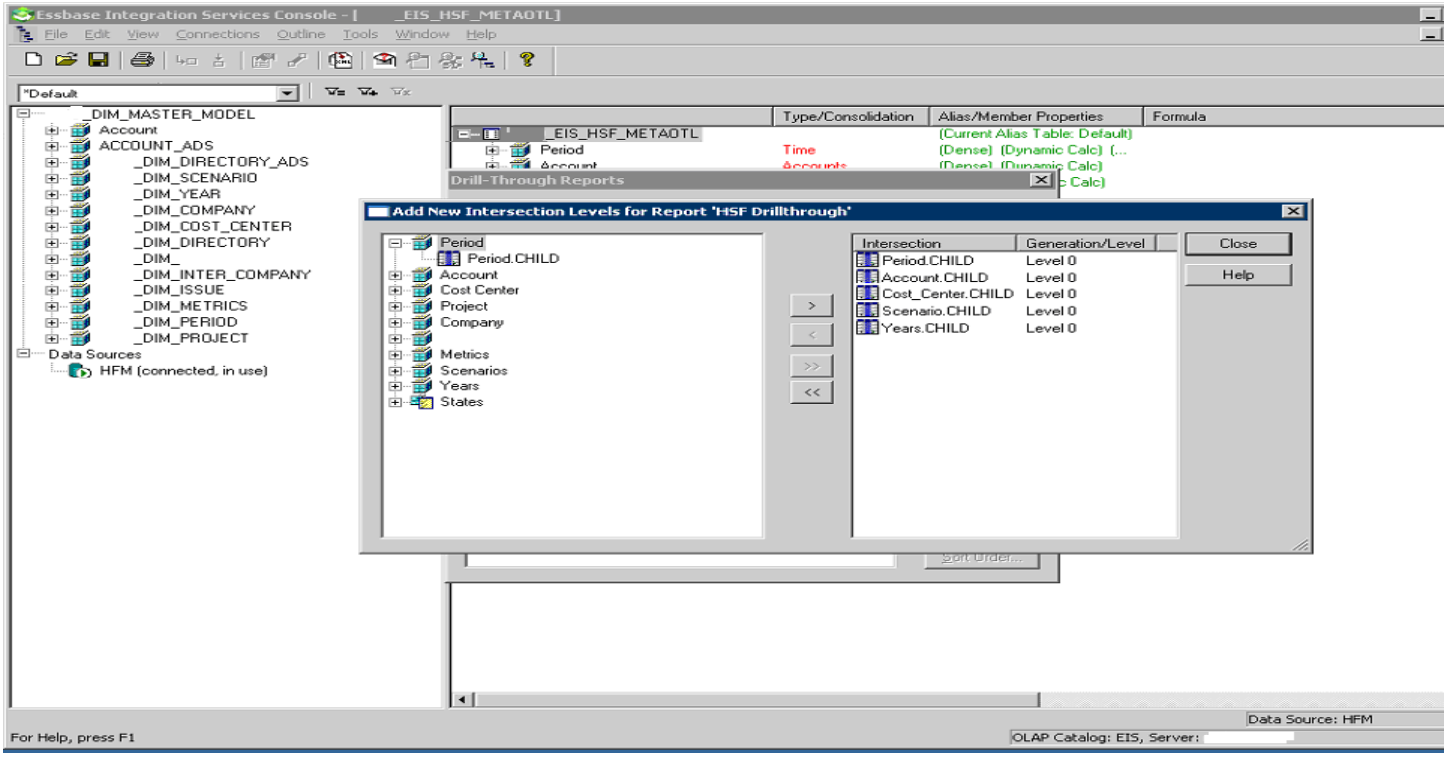
E. Once logged in we need to select Edit→ Drill-Through Reports from the main menu.



F. Next we will need to select the Drill-Through report and click on Template SQL. The Drill-Through Report will have the OLAP Intersection levels and the report columns. Since the screen shot is of an existing report we are seeing the Intersection levels, while creating a new report we will have to click on Add and select the respective levels which need to be added to create the report.

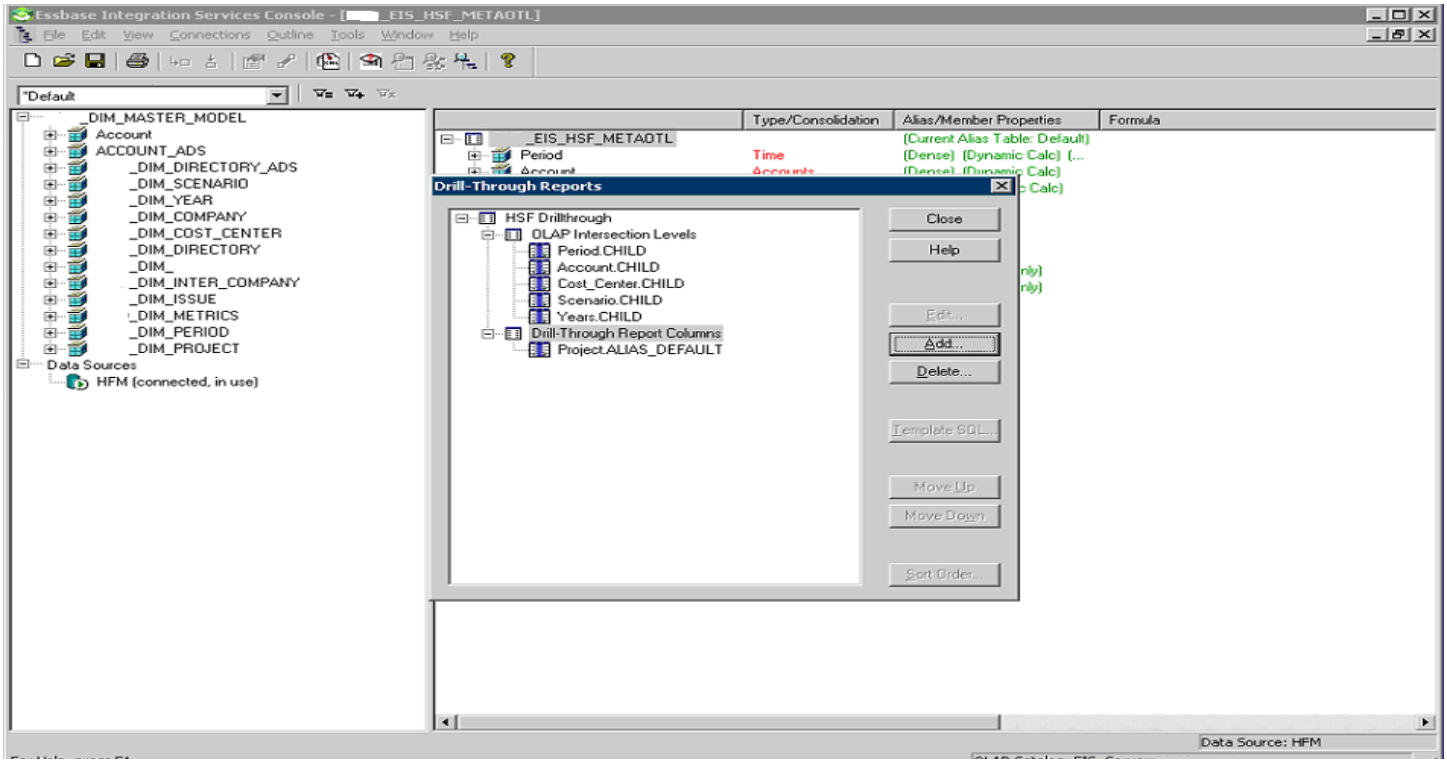


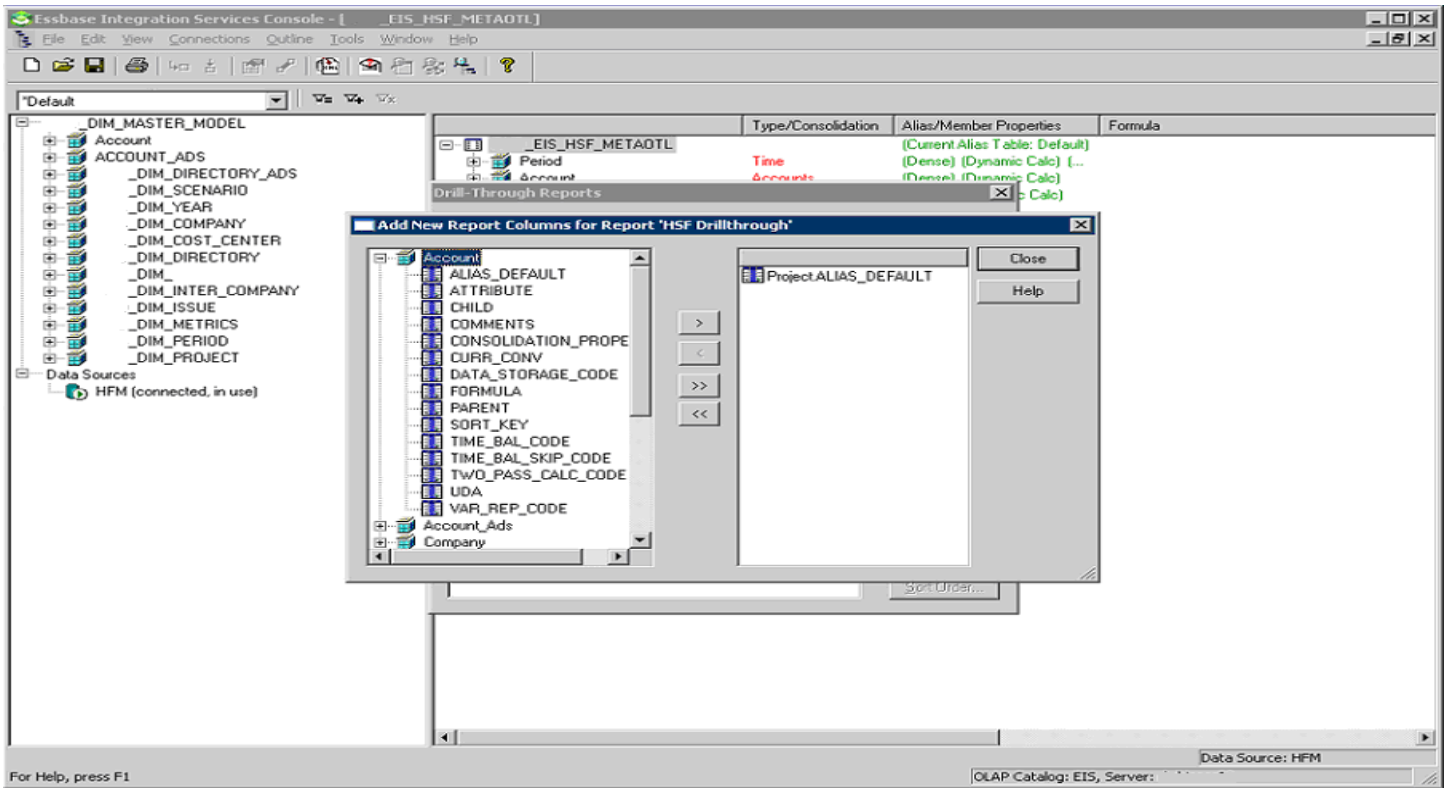
G. In order to define the OLAP intersection levels we need to select all the level 0 members.

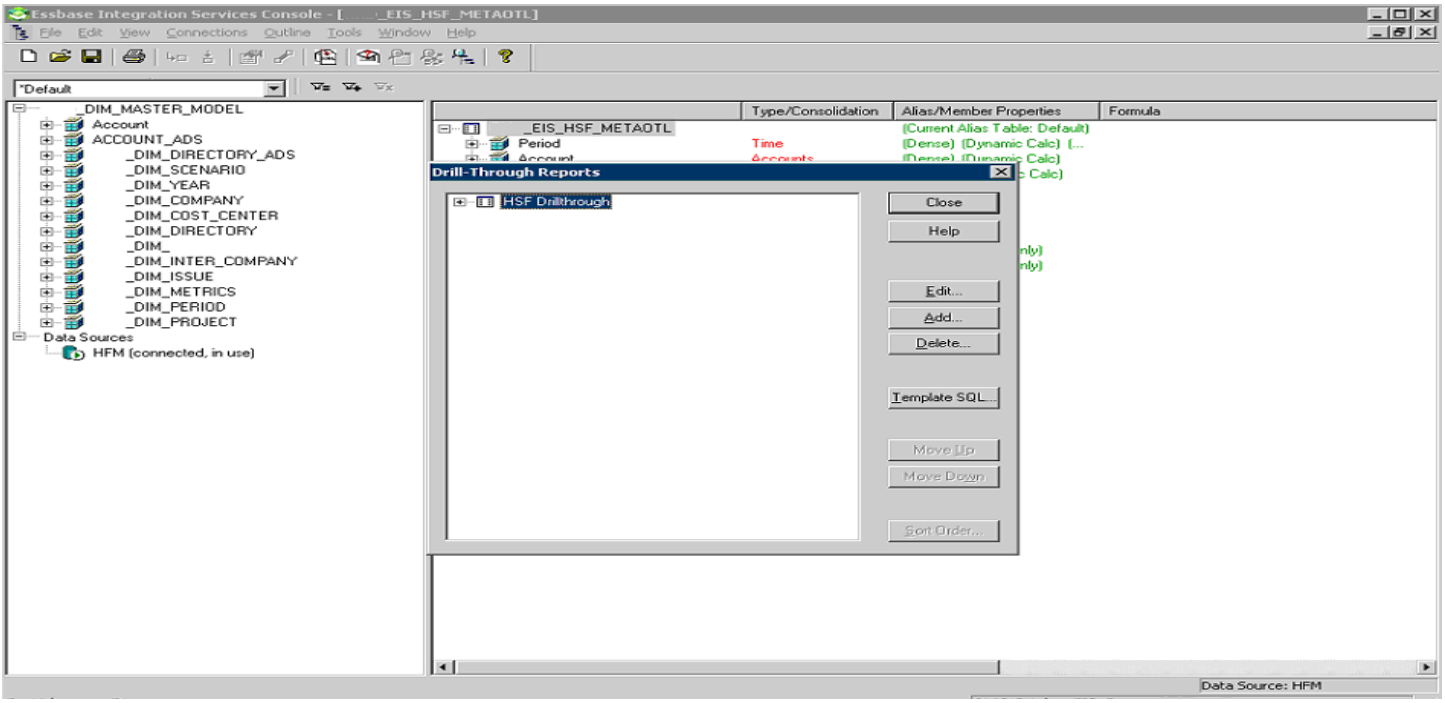


All those columns on which data needs to be filtered should be added here and we need to select the child members and the members should be at Level 0.

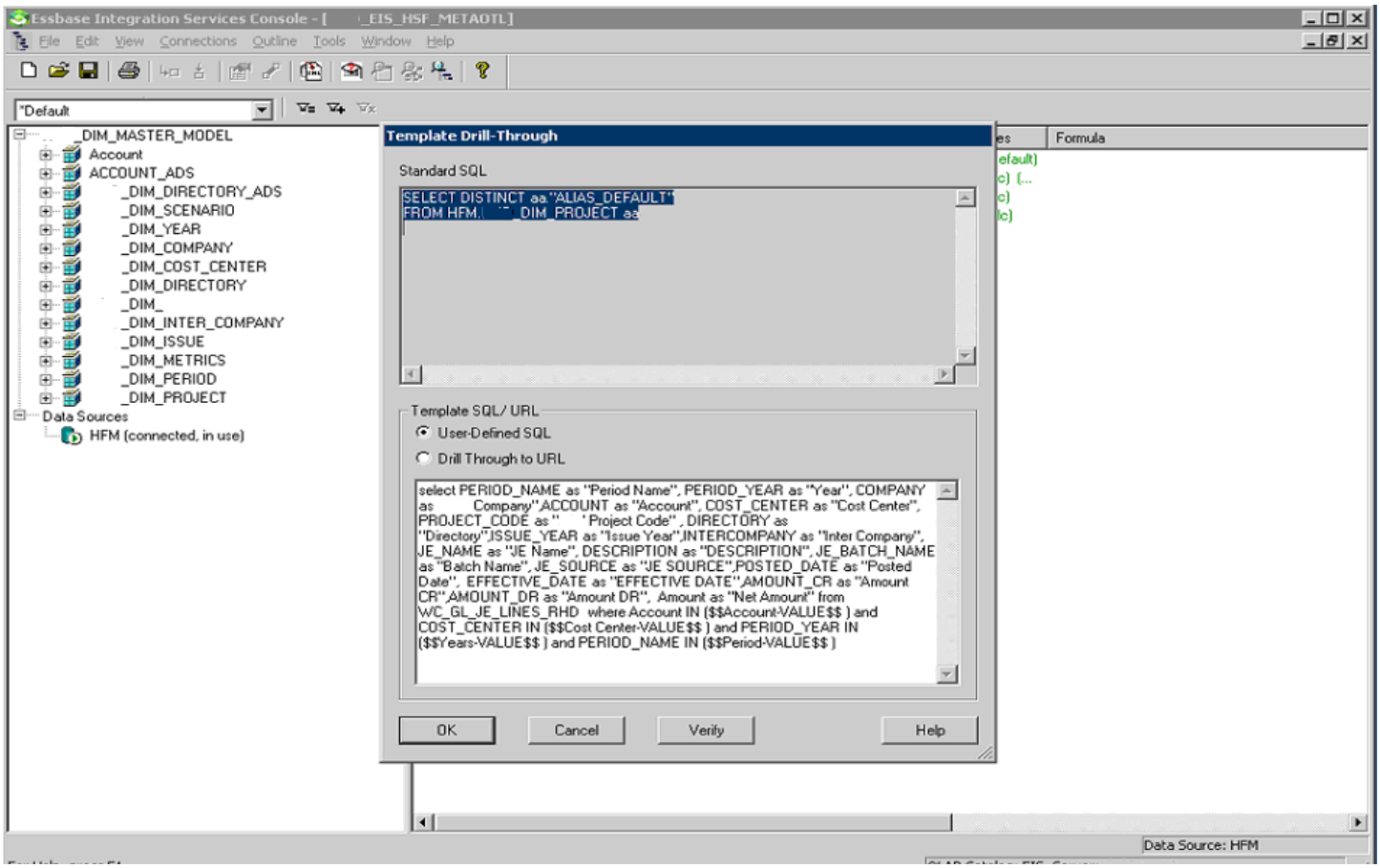
H. Next we can also add Drill-Through Report columns. In case we are using columns from the existing OLAP model itself then we can use this option. All the required columns available in the schema can be added here as visible in the screen shot below. But since we are using the Template SQL for the query we don't need to set this up.





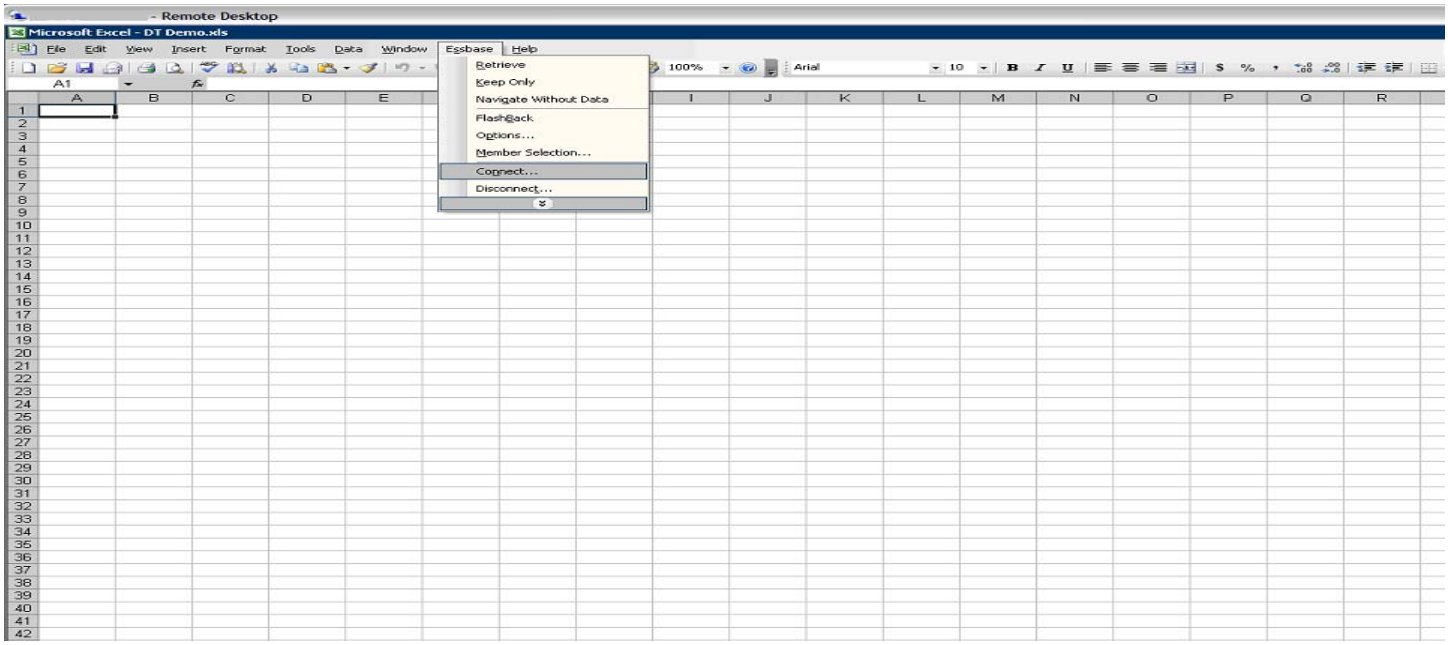


I. Next we can select "Template SQL" to provide the custom SQL Query which needs to be used for drill through.

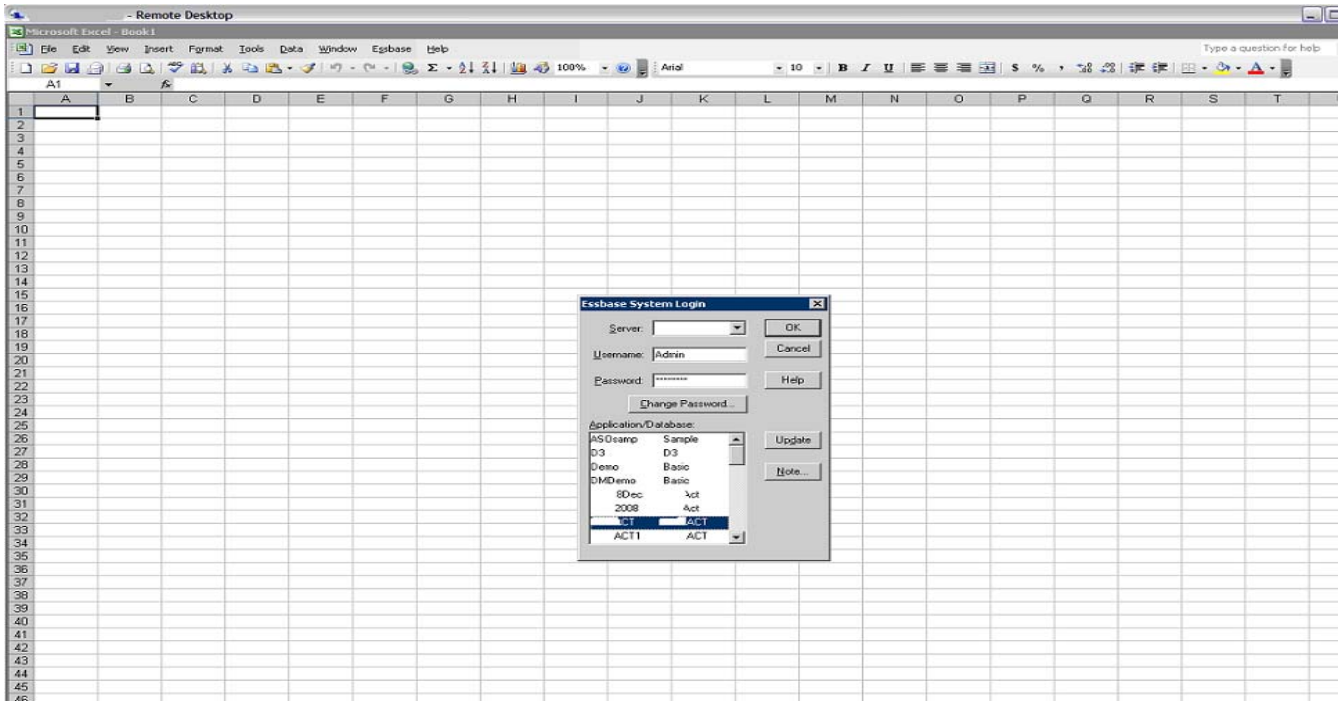


J. We can also verify the query by clicking on Verify button as visible above and if the query is correct then we get a window with a success message.

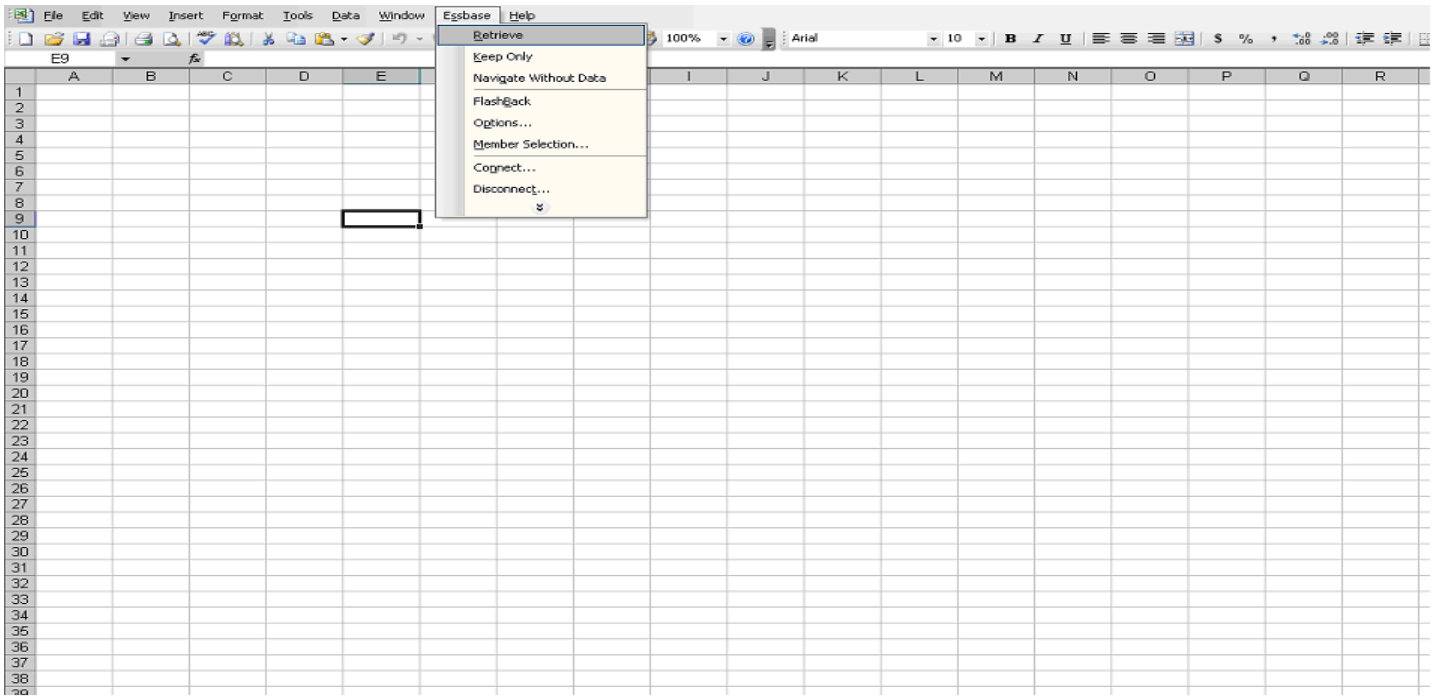
K. This finishes the setup part of the Drill-Through Reporting, now to check the solution we need to login to MS-Excel which should have Essbase Connector installed. As visible on the screen-shot we will get a Essbase System Login screen on clicking on the Essbase→ Connect option in the main menu.



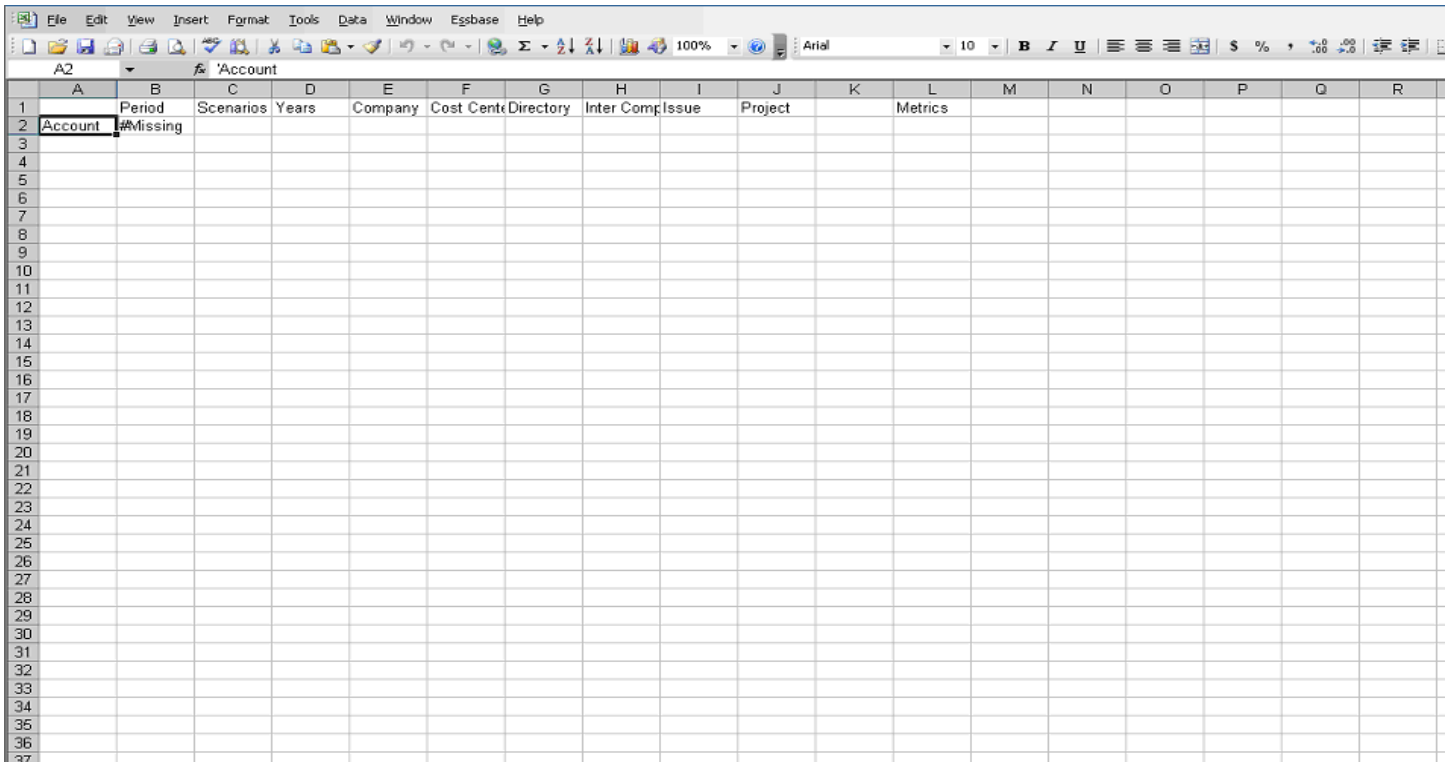
L. This will open the Essbase System Login window where we need to provide the Username and Password and click Ok. This will display a list of Application/Databases, please select the required one and click Ok.



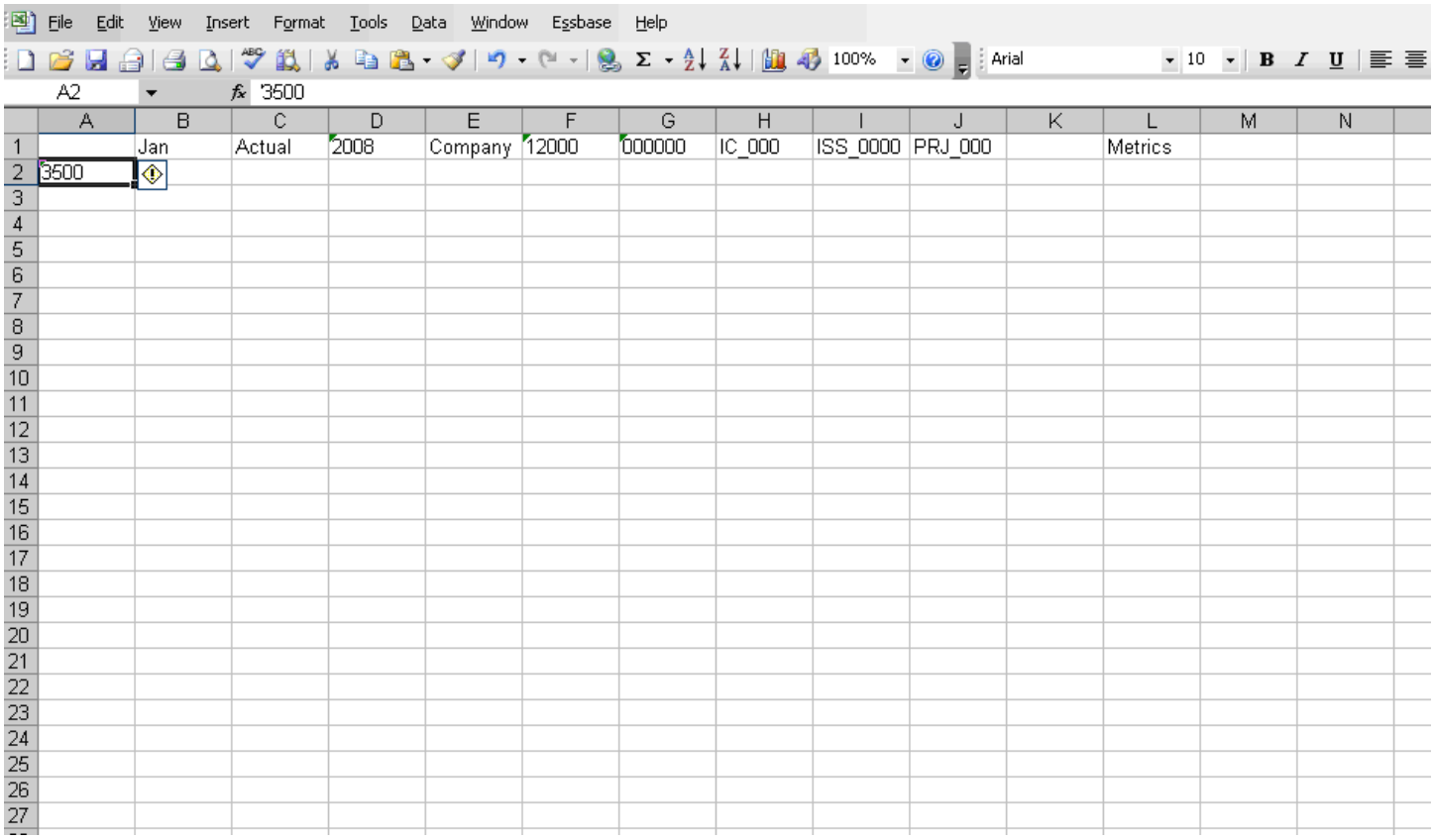
M. Next we need to select Essbase → Retrive.



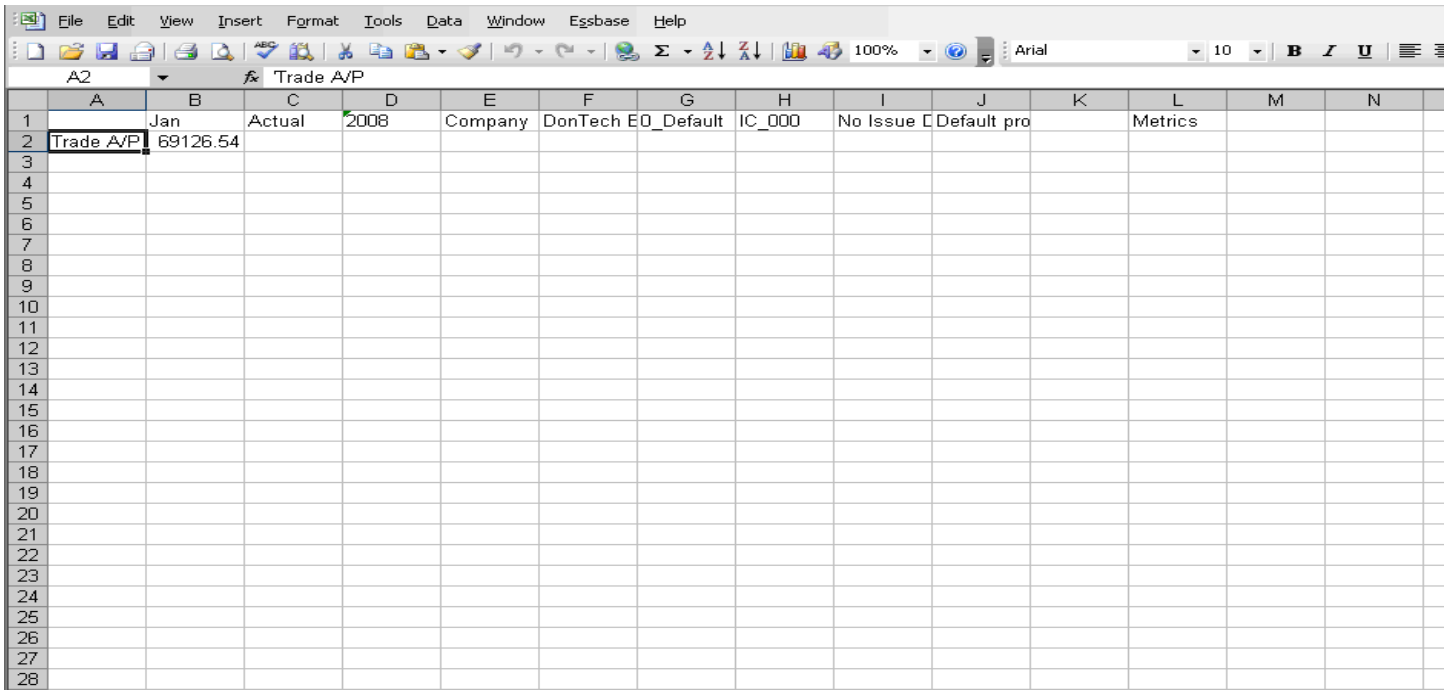
N. This will show all the required dimensions.



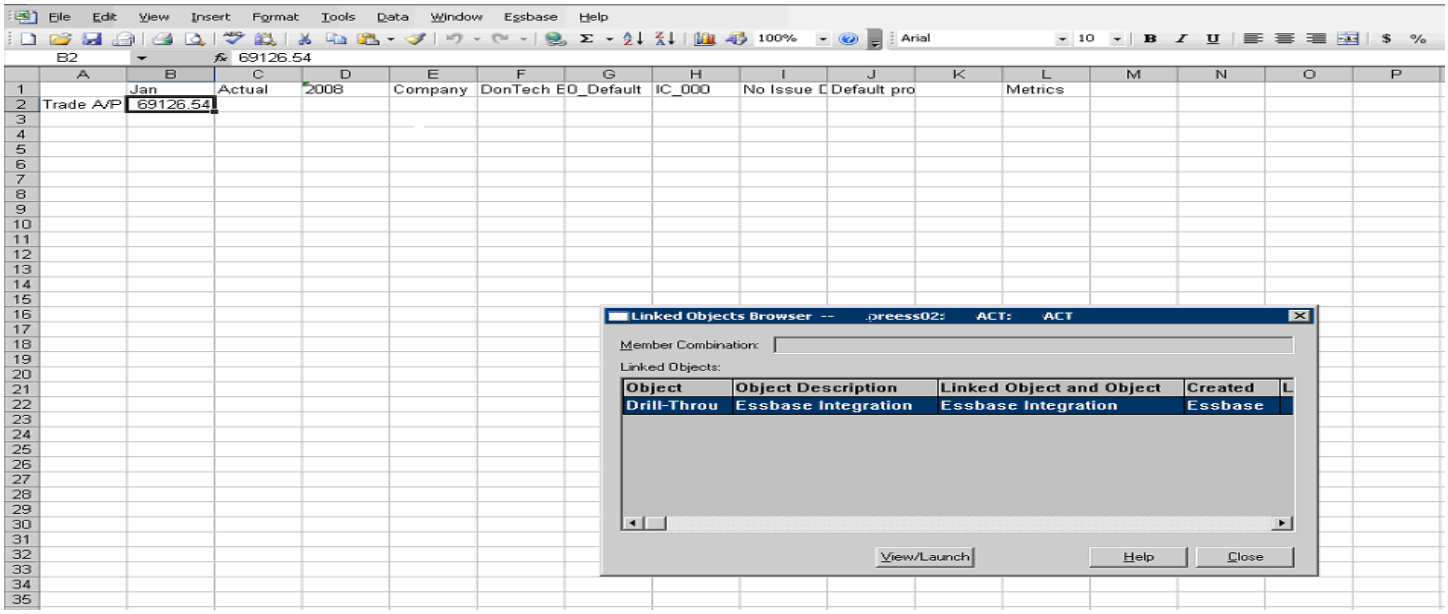
O. Next we need to provide the required dimension values to drill through on the selected data. We need to be on level zero on the required dimensions. The below screen shot will show an example.



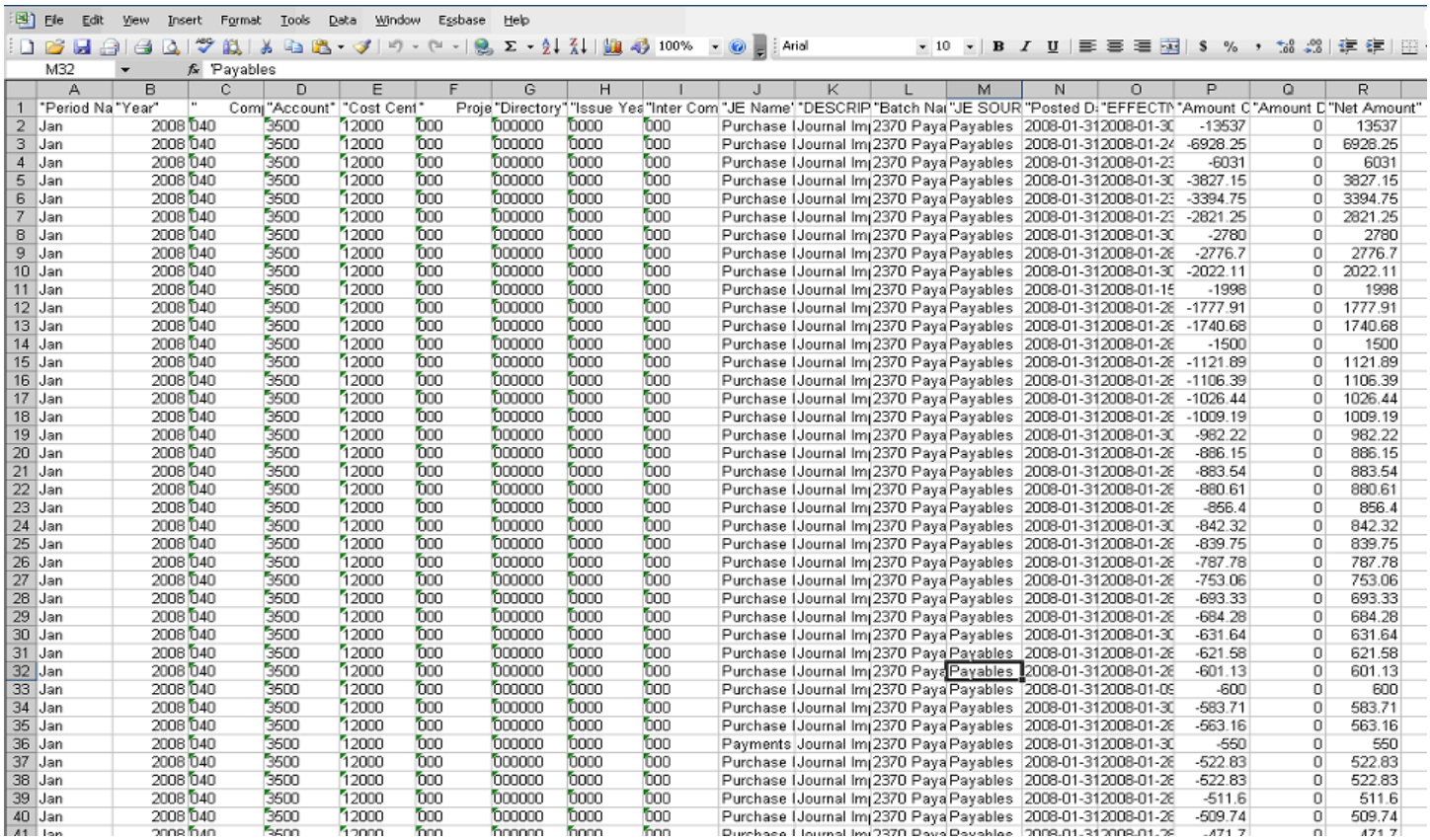
P. If we double click on the given Account then it will give the balance for the given period.



Q. Next in order to view the drill through data we need to double click on the balance amount after which we need to select the Drill-Through report and click on View/Launch.



R. Clicking on View/Launch will open an excel sheet with the detailed entries for the above balance. This will open one more sheet and the detailed data will be available in this sheet.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	"Period Na	"Year"	"	Com	"Account"	"Cost Cent	"	Proje	"Directory"	"Issue Yea	"Inter Com	"JE Name"	"DESCRIP	"Batch Na	"JE SOUR	"Posted D	"EFFECTI	"Amount C	"Amount D	"Net Amount"
2	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-13537	0	13537
3	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-24	-6928.25	0	6928.25
4	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-23	-6031	0	6031
5	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-3827.15	0	3827.15
6	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-23	-3394.75	0	3394.75
7	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-23	-2821.25	0	2821.25
8	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-2780	0	2780
9	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-2776.7	0	2776.7
10	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-2022.11	0	2022.11
11	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-15	-1998	0	1998
12	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-1777.91	0	1777.91
13	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-1740.68	0	1740.68
14	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-1500	0	1500
15	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-1121.89	0	1121.89
16	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-1106.39	0	1106.39
17	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-1026.44	0	1026.44
18	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-1009.19	0	1009.19
19	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-982.22	0	982.22
20	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-886.15	0	886.15
21	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-883.54	0	883.54
22	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-880.61	0	880.61
23	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-856.4	0	856.4
24	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-842.32	0	842.32
25	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-839.75	0	839.75
26	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-787.78	0	787.78
27	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-753.06	0	753.06
28	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-693.33	0	693.33
29	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-684.28	0	684.28
30	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-631.64	0	631.64
31	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-621.58	0	621.58
32	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-601.13	0	601.13
33	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-09	-600	0	600
34	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-583.71	0	583.71
35	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-563.16	0	563.16
36	Jan	2008	040		3500	12000	000		000000	0000	000	Payments	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-30	-550	0	550
37	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-522.83	0	522.83
38	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-522.83	0	522.83
39	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-511.6	0	511.6
40	Jan	2008	040		3500	12000	000		000000	0000	000	Purchase I	Journal Im	2370 Paya	Payables	2008-01-31	2008-01-28	-509.74	0	509.74

This as such finishes the Drill-Through Demo and the solution.

In the end I would just like to summarize as to in which situations/scenarios we can use this solution approach.

The following scenarios can be considered for using this solution:

1. Short implementation time frame with quick turnaround time.
2. Very less or minimum customization requirement in OBIA.
3. Budgetary constraints with very little implementation/customization expenses.
4. The Drill-Through solution can also be implemented if we don't have a warehouse as it can directly allow you to fetch data from the source system with the Template SQL option.
5. There is existing setup of OBIA & Hyperion and we need to leverage them to provide a optimal solution.