Microsoft Business Intelligence for Oracle PeopleSoft Enterprise Management (EPM)

**White Paper FOR Solutions PARTNERS**

Dated: December 5, 2007

Updated: March 7, 2008

Theinformation contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, Excel, PerformancePoint, SharePoint, Visio, Visual Basic, Visual Studio, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

[Executive Overview 4](#_Toc192494133)

[PeopleSoft EPM Overview 4](#_Toc192494134)

[The Challenge Posed by Oracle and Fusion Intelligence 6](#_Toc192494135)

[Microsoft BI Solution 6](#_Toc192494136)

[Example Opportunity 7](#_Toc192494137)

[Microsoft BI Architecture for Peoplesoft 8](#_Toc192494138)

[Microsoft Office PerformancePoint Server Business Value 9](#_Toc192494139)

[PeopleSoft Core Concepts 11](#_Toc192494140)

[Understanding PeopleSoft Tables 11](#_Toc192494141)

[Understanding PeopleSoft Meta Data 11](#_Toc192494142)

[Understanding Trees 12](#_Toc192494143)

[Impact and Correct Use of SETIDs and Effective Dates 15](#_Toc192494144)

[The PeopleSoft Financials Data Mart 16](#_Toc192494145)

[Model Overview 17](#_Toc192494146)

[PeopleSoft Enterprise 8.8 Financials Solution Demonstrator 17](#_Toc192494147)

[Installation 17](#_Toc192494148)

[Integration Services Packages 19](#_Toc192494149)

[Package Customization 19](#_Toc192494150)

[Package Execution 19](#_Toc192494151)

[List of Source Tables 20](#_Toc192494152)

[Analysis Services Cube 22](#_Toc192494153)

[Reporting Services Reports 22](#_Toc192494154)

[PerformancePoint Server Dashboards 23](#_Toc192494155)

[PerformancePoint Server Planning Application 24](#_Toc192494156)

[Summary and Resources 25](#_Toc192494157)

[Appendix 27](#_Toc192494158)

# Executive Overview

According to Gartner, Microsoft Corporation has been positioned in the Leaders Quadrant in the “[Magic Quadrant for Business Intelligence Platforms, 2008](http://mediaproducts.gartner.com/reprints/microsoft/vol7/article3/article3.html)” report.

**About the Magic Quadrant**

Gartner, Inc. "Magic Quadrant for Business Intelligence Platforms, 2008" by James Richardson et al, February 1, 2008. The Magic Quadrants are copyrighted 2007 and 2008, by Gartner, Inc. and is reused with permission. The Magic Quadrants are a graphical representation of a marketplace at and for a specific time period. It depicts Gartner’s analysis of how certain vendors measure against criteria for that marketplace, as defined by Gartner. Gartner does not endorse any vendor, product or service depicted in the Magic Quadrant, and does not advise technology users to select only those vendors placed in the “Leaders” quadrant. The Magic Quadrant is intended solely as a research tool, and is not meant to be a specific guide to action. Gartner disclaims all warranties, express or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

Business intelligence (BI) and performance management (PM) solutions based on the Microsoft platform help to solidify Microsoft SQL Server as one of the database platforms of choice for PeopleSoft, while still influencing executive users of BI and PM solutions as to the value of Microsoft. This white paper is intended to provide Microsoft solutions partners with information on a Microsoft business intelligence solution for Oracle PeopleSoft customers who run PeopleSoft on Microsoft® SQL Server®. In addition, it provides an overview of particular PeopleSoft concepts which are particularly important when building a business intelligence solution. The purpose of the solution illustrated in this white paper, and the accompanying solution demonstration, is to show PeopleSoft customers how:

1. A Microsoft BI solution can meet both BI and PM requirements in a single integrated solution; and
2. How Microsoft understands the technical issues presented by PeopleSoft and provides a robust solution for each issue.

According to [Oracle](http://www.oracle.com/technology/deploy/performance/pdf/oracle_for_peoplesoft.pdf) , 24% of PeopleSoft applications utilize Microsoft SQL Server (Oracle represents 69% and IBM DB2 7%). Microsoft believes that Oracle has two objectives with PeopleSoft/SQL customers: 1) migrate them from SQL Server to Oracle, and 2) align them with the Fusion Applications roadmap. Both of these objectives are needed to ensure customers remain Oracle customers and do not migrate their application layers to a non-Oracle solution provider. By also helping customers move from Microsoft Windows Server® to the Linux operating system, Oracle further solidifies their position as the application and platform provider for these customers.

Microsoft and Microsoft solutions partners can retain PeopleSoft/SQL customers, and grow revenue, by offering additional value to the Microsoft platform—particularly on the SQL Server platform. PeopleSoft customers who run PeopleSoft on SQL Server may not be well served by business intelligence solutions from Oracle. These customers must usually turn to a third-party solution, which previously would have been most likely Cognos (now owned by IBM) and Business Objects (now owned by SAP). Therefore a rather unique opportunity exists to take advantage of this confusion and absence of solution to educate Microsoft customers as to how a Microsoft BI solution for PeopleSoft can help protect them from changes in Oracle’s BI strategy, the consolidation that is happening in the BI market segment, and also help deliver a BI solution which utilizes their investment and skills in SQL Server and other Microsoft technologies.

## PeopleSoft EPM Overview

The Oracle PeopleSoft Enterprise Performance Management (EPM) solution allows businesses to obtain exceptional performance by matching up the critical information and resources with critical business goals. PeopleSoft EPM provides performance-management software for all budgets and phases of the management cycle, allowing managers to create strategies to improve growth, tie in strategies with business value plans, and track ongoing operations in an effective manner.

The PeopleSoft Enterprise Warehouse includes the following:

* A business analytics platform that is already setup to provide automated data integration, staging, transformation, enrichment, and multi-dimensional reporting and analysis.
* A pure-Internet administration layer that can be used easily to provide data management, enrichment, scheduling, and security.
* Pure Internet tabular, pictorial, and unplanned reporting and analysis software with many different report template definitions.
* An extensive business data model that details a company’s financial, personnel, customer, and vendor information.
* Special communication that allows a business to include analytics in their ongoing business processes.
* Adaptable and scalable open architecture that is engineered to perform and scale.
* An Operational Data Store (ODS) that includes staging and reporting areas to create virtually instantaneous operational reports and exploration.
* Embedded meta data features that can assist with the improvement of a business’s data and applications.
* A fast implementation which is accomplished via a predetermined enterprise warehouse model.
* Simple and effective data integration is provided for PeopleSoft and non-PeopleSoft systems, along with third-party data providers.
* Popular ETL technology from Informatica that includes an estimated 1,200+ predefined ETL maps and integrated meta data.
* A data mart builder that creates relational data marts that can be used with multidimensional investigations.
* Predetermined data marts that are engineered to assist with role-, function-, and job-specific investigations with an open reporting strategy that allows a business to use popular online analytical processing (OLAP) and reporting technology.
* Simple integration with online transactional processing (OLTP) systems, older systems, and third-party data content providers.

Put in simple terms, PeopleSoft EPM is a data warehousing solution for PeopleSoft Enterprise applications, including pre-built data models, pre-built extract, transform, and load (ETL) routines, and pre-built analytics, reports, key performance indicators (KPIs), and scorecards. Some pre-built analytic applications are also included, such as Workforce Planning, Workforce Rewards, and Funds Transfer Pricing. The modules of PeopleSoft EPM are listed below:

* CRM Analytics
* Financial Analytics
* Industry Analytics
* Supply Chain Analytics
* Workforce Analytics
* Performance Management Warehouse

In the latest release (August 2006) , PeopleSoft EPM 9, Oracle announced support for JDE World and OneWorld customers, and also talked about integration of non-Oracle data, such as SAP.

On the surface PeopleSoft EPM 9 would seem to be a compelling solution for PeopleSoft customers, an integrated BI and CPM solution that fits exactly the data and needs of a PeopleSoft customer. However, on examination of the chronology of the PeopleSoft EPM solution, you can see exactly why some customers and independent observers are left confused as to Oracle’s direction and intention for this solution.

1999: PeopleSoft, Inc. announces availability of EPM and partnership with Informatica for ETL.

2002: Cognos Inc. announces support in Series 7 for PeopleSoft EPM 8.3.

2003: Business Objects announces PeopleSoft, Inc. to re-sell Business Objects Enterprise 6 for PeopleSoft EPM.

2004: PeopleSoft, Inc. switches from Informatica to Ascential Software for ETL, ending 5 year relationship with Informatica.

2004: Oracle buys PeopleSoft, Inc.

2005: IBM buys Ascential Software.

2006: Oracle Corporation announces Fusion Intelligence and migration path for PeopleSoft EPM; then later drops Fusion Intelligence in favor of a Fusion-BI solution based on the Siebel Analytics platform (see next acquisition).

2006: Oracle buys Siebel Systems, Inc. including Siebel Analytics.

2007: Oracle buys Hyperion Solutions Corporation, which also offers enterprise performance management solutions.

2008: SAP buys Business Objects.

2008: IBM buys Cognos.

In summary, there is now an Oracle-owned solution which has remained relatively constant in terms of data model, but now consists of ETL technology provided by IBM Corporation and BI tools technology soon to be provided by SAP. Microsoft believes that Oracle’s approach is to sell consulting services and Oracle Business Intelligence Enterprise Edition (or Standard/Standard One); which may include using Oracle Warehouse Builder for ETL. This strategy therefore negates much of the benefit associated with a pre-built EPM solution, and does little for a customer running PeopleSoft on SQL Server.

Note: Versions of PeopleTools (for PeopleSoft EPM) contain a feature called ”Cube Manager.” This feature enables the automatic generation of Microsoft SQL Server Analysis Services cubes (in addition to Informatica, Essbase, and Cognos PowerPlay). The focus of this capability was Essbase and PowerPlay and these remain the only platforms listed and described in the Oracle documentation. It is unclear as to whether Analysis Services remains as a supported platform (especially with SQL Server 2005).

## The Challenge Posed by Oracle and Fusion Intelligence

As Oracle migrates capabilities of PeopleSoft Enterprise, J D Edwards, and Siebel towards Fusion Applications, so too does it migrate the legacy BI and reporting capabilities of those applications into a new Fusion Middleware Business Intelligence solution for Fusion Applications. The basis for this BI solution are the BI products Oracle acquired with Siebel Systems (which itself is the product of an acquisition of a technology called nQuire by Siebel Systems). It is well known that Fusion Applications will only run on the Oracle database platform. Unless Microsoft and its solutions partners take action now to build the customer’s perception of the value of Microsoft, there is a high probability that Microsoft will lose customers/application revenues.

# Microsoft BI Solution

The Microsoft BI solution for PeopleSoft utilizes all of the features of SQL Server, Office PerformancePoint® Server, Microsoft Office Excel®, and Office SharePoint® Server to deliver a compelling BI and CPM experience for PeopleSoft customers. As listed previously, the target customer is a current Oracle PeopleSoft Enterprise customer who intends to continue to run PeopleSoft for the foreseeable future. The customer ideally runs PeopleSoft on SQL Server (likely version 2000 or 2005). The customer has not implemented PeopleSoft EPM, or is dissatisfied with PeopleSoft EPM and/or the customer requires a planning and budgeting solution and has not implemented or is dissatisfied with the PeopleSoft Planning and Budgeting solution.

Additionally, the customer may be happy with PeopleSoft EPM Performance Management Warehouse, and their chosen ETL tool, but may be unhappy with the BI tools used to provide the experience to business users.

A typical PeopleSoft EPM customer or customer who needs a BI or Planning and Budgeting solution and who does not have PeopleSoft EPM, may express the following needs:

* Wants to publish information in a dashboard.
* Wants to build reports on PeopleSoft data (and maybe include non-PeopleSoft data).
* Wants an Office Excel-based planning and budgeting solution (as opposed to a Web interface).
* Wants an environment in which to perform ad-hoc analysis of financial and non-financial (drill-down/up, drill-by, drill-through) data.
* Wants to make PeopleSoft data available through Office SharePoint Server.
* Wants to allow their users to use Office Excel for financial data analysis and reporting, and enable this content to be published and shared easily.

Microsoft’s BI solution can meet these needs in a solution that includes products that work well together. In addition, Microsoft understands the technical issues presented by PeopleSoft, and the solution illustrated in this paper (along with the solution demonstration) will assist in showing how Microsoft addresses these issues.

## Example Opportunity

The target customer is a current Oracle PeopleSoft Enterprise customer who intends to continue to run PeopleSoft for the foreseeable future. The customer ideally runs PeopleSoft on SQL Server (likely version 2000 or 2005). The customer has not implemented PeopleSoft EPM or is dissatisfied with PeopleSoft EPM and/or the customer requires a planning and budgeting solution and has not implemented or is dissatisfied with the PeopleSoft Planning and Budgeting solution. The demonstration solution discussed in this white paper will have maximum impact on those customers running PeopleSoft Financials, though the concepts apply to all PeopleSoft Enterprise applications.

License revenue opportunity:

1. Up sell from SQL Server 2000 to SQL Server 2005.
2. Sell additional SQL Servers and SQL Server client access licenses (CALs) or SQL Server processors for BI workload.
3. Sell Microsoft Office PerformancePoint Server 2007.
4. Sell Microsoft Office SharePoint Server 2007 and Enterprise CAL (ECAL).

Service revenue opportunity:

1. Sell Microsoft solutions-partner solution delivery.

**Example license revenue opportunity (approximation):**

Customer is running PeopleSoft Enterprise on one (1) SQL Server 2000 Enterprise with 200 SQL Server CALs; no software assurance (SA). The following costs are an approximation ONLY and are based on Select-C estimated retail prices. Software Assurance is not included, which would be another 25% per year. Exact pricing should be obtained from solutions partner’s large account reseller (LAR). For more information about licensing opportunities read “[Volume Licensing Overview](http://www.microsoft.com/licensing/resources/default.mspx).”

|  |  |  |  |
| --- | --- | --- | --- |
| **Product** | **Quantity** | **Select-C Estimated Retail Prices (U.S.D.)** | **Total** |
| SQL CAL 2005 Device CAL | 200 | $                    138.00 | $    27,600.00 |
| SQL Server Enterprise Edition 2005 IA64 | 3 | $                 7,197.00 | $    21,591.00 |
| Office PerformancePoint Server 2007 | 1 | $              16,959.00 | $    16,959.00 |
| Office PerformancePoint CAL 2007 Device CAL | 200 | $                    166.00 | $    33,200.00 |
| Office SharePoint Server 2007 | 1 | $                 3,751.00 | $      3,751.00 |
| Office SharePoint CAL 2007 Device CAL | 200 | $                       64.92 | $    12,984.00 |
| SharePoint Enterprise CAL 2007 Device CAL | 200 | $                       64.00 | $    12,800.00 |
|  |  |  | $  128,885.00 |

# Microsoft BI Architecture for Peoplesoft

Figure 1. Depicts the high-level architecture and products typically deployed as part of a Microsoft BI solution for PeopleSoft. The physical deployment architecture may differ according to scale and other factors which influence the operating environment; for example the type of storage system that is used.



Windows client

Performance-

Point Server 2007 Planning

PerformancePoint Server 2007 Plan-ning Staging DB

SQL Server

**Figure 1.**

Note: Management tools and Business Intelligence Development Studio are not shown.

As can be seen in Figure 1; the solution is a fairly typical data warehousing or data mart solution. Data is extracted from PeopleSoft database tables using a native database connection from within SQL Server Integration Services. The frequency of the data load is defined by the customer, for example intra-day, daily, weekly, or monthly. This extracted data is then loaded into a target dimensional model, which is then further transformed through a series of database views to act as a data source for SQL Server Analysis Services. A series of SQL Server Reporting Services reports, ProClarity briefing books and Office PerformancePoint Server dashboards, scorecards, and analytics are then used to present this data, hosted within Microsoft Office SharePoint Server. Finally data is exported into an Office PerformancePoint Server Planning staging database which is then used to load the Office PerformancePoint Server Planning system and therefore support performance-management processes around PeopleSoft ledger and budget ledger data.

Additional Note: An alternative architecture to the above option would be to replace the “PeopleSoft Financials Data Mart SQL Server” with the Oracle PeopleSoft Performance Management Warehouse (core of PeopleSoft EPM), when the customer has implemented this solution and is happy with what it provides. In this type of architecture, the Microsoft BI and CPM solution would operate with the existing PeopleSoft Performance Management Warehouse.

## Microsoft Office PerformancePoint Server Business Value

Microsoft Office PerformancePoint Server 2007 is an integrated performance management application designed to help improve operational and financial performance across all departments and all levels of your organization. With Office PerformancePoint Server 2007, you can monitor progress, analyze what is driving variances, and plan your business from budgeting to creating management reports. You can have metrics, key performance indicators (KPIs), and reports delivered to every desktop through intuitive scorecards, dashboards, and the easy-to-use Microsoft Office system environment. A key component of the Microsoft Business Intelligence (BI) offering, Office PerformancePoint Server 2007 can help you understand how performance can align with personal and departmental goals and objectives.

The ability to monitor and measure results is a key component of running a successful business. Microsoft Office PerformancePoint Server 2007 delivers the next generation of Business Scorecard Manager, which can help you define and use scorecards and key performance indicators (KPIs) to drive accountability and alignment across your organization. With its rich visualization, user-friendly design, and familiar Microsoft Office system environment, Office PerformancePoint Server 2007 can help all users easily build, personalize, manage, and use scorecards. Its rich monitoring capabilities:

* Span from personal performance dashboardsto formal methodology-based and cascading scorecards;
* access a wide range of structured and unstructured data-sources for more comprehensive dashboards and scorecards;
* include rich dashboard filters to dynamically filter all views on the dashboard;
* include contextual KPIs and reports that reflect changes in planning, budgeting, and forecasting data;
* include DynamicStrategy Mapslinked to KPI data; and
* assign owners to KPIs for accountability.

The Office PerformancePoint Server Dashboard Designer has a familiar Microsoft Office 2007 look and feel that provides:

* Fast time to delivery and ease of administration;
* dashboards that are designed for collaboration and reuse;
* offline capabilities; and
* built-in templates and wizards that allow users to quickly build and share dashboards and scorecards.

Business users often depend on IT experts to provide them with the business intelligence (BI) data they need to make decisions. The Microsoft BI offering includes Microsoft Office PerformancePoint Server 2007 Analytics, built from the industry-leading analytic solution from ProClarity. It can help your organization go beyond analyzing data to capturing and sharing analytical best practices using the intuitive, collaborative environment of the 2007 Microsoft Office system.

http://www.microsoft.com/bi/images/clearpixel.gif

With Office PerformancePoint Server 2007, employees can exceed standard BI analysis by creating analytics that include user-driven, centrally managed key performance indicators (KPIs) and interactive visualization tools such as the patented ProClarity decomposition tree. These rich analytic applications can help users rapidly identify trends, opportunities, and even threats hidden within large quantities of data. This analytics and data visualization includes:

* Web-based thin analysisthatprovides users with rich analytical functionality “out-of-the-box,” minimizing the help required from IT staff;
* analytics that provide multidimensional slice and dice, drill-across, drill-to-detail, root-cause analysis, and centralized business logicdefinitions;
* predictive analyticsthatallow for forecasting and trend analysis;
* guided and contextual analysisthrough tight integration of KPIs and analytics in the dashboard; and
* advanced data visualizationssuch as decomposition trees and performance maps.

Built on the stability, security, and scalability of Microsoft SQL Server 2005 and the Microsoft Business Intelligence (BI) platform, Microsoft Office PerformancePoint Server 2007 can help your business transform planning, forecasting, and budgeting into effective and dynamic performance management. Office PerformancePoint Server 2007 enables your organization to interact with and continuously contribute to business planning, budgeting, and forecasting—again, through the easy-to-use Microsoft Office system environment. Its Planning and Modeling capabilities:

* + Allow budget administrators to design models the way they think about their business;
  + include top-down and bottom-up planning functionality that connects people and drives accountability;
  + include process management to manage forms, workflows, submissions, approvals, reports, notifications, and annotations;
  + feature a highly-secure and auditable environment; and
  + span from single model deployments to enterprise scenarios.

Microsoft Office PerformancePoint Server 2007 Reporting and Consolidation capabilities:

* + allow users to construct models that drive profitability analysis incorporating standard financial functionality;
  + support management and GAAP consolidation process with multiple currency conversions,   
    inter-company eliminations and reconciliations, and multi-tier allocations;
  + provide dynamic and standard reports that include financial and business performance reports; and
  + can publish live reports from Excel to Reporting Services and Microsoft Office Server providing consistency of experience for report consumers.

For more information about Microsoft Office PerformancePoint Server 2007, please read the “[Office PerformancePoint Server Resources](#PerformancePoint_Resources)” section.

## PeopleSoft Core Concepts

The following section will aid the understanding of particular PeopleSoft concepts (usually implemented at the database layer) which are particularly important when building a business intelligence solution. The aim of this section is not to provide a comprehensive description of the concepts listed and their application in the PeopleSoft applications, rather it is to give a basic understanding of these concepts and why they are important for analysis and reporting scenarios.

### Understanding PeopleSoft Tables

The PeopleSoft application database is a typical “operational database” or online transaction processing (OLTP) database in that it aims to store data only once and enable efficient processes for the creation, read, update, and deletion of records through the PeopleSoft application. Each application (e.g. Financials and Supply Chain Management or Human Resources Management Systems [HRMS]) therefore contains many thousands of tables. Understanding these tables and the role played by the different types of tables is therefore important when building a business intelligence solution.

#### Base Tables

A base table stores live or operational data which is constantly changing. In Financials, an example of a base table is PS\_LEDGER. This table is a core table of the PeopleSoft General Ledger and contains period activity according to the various chartfields, some of which are fixed and some of which can be customized to meet the needs of the organization.

#### Control Tables

A control table contains a list of values which is used to qualify data in a base table. For example, the table PS\_LEDGER contains the period activity by Account, whereas the table PS\_GL\_ACCOUNT\_TBL contains the list of Accounts. Control tables are normally identified by their post-fix ”\_TBL.” Another name for a control table is a ”lookup table.”

#### Application Tables

PeopleSoft stores application rules and data in application tables. Application tables are normally identified by their use of the prefix ”PS” without the following ”\_” character (e.g. PSTREENODE).

Other table types include “Views” and “Reporting Tables,” however, as these types are not used or referenced by the solution demonstrator, it will simplify this section to ignore these types for the time being.

### Understanding PeopleSoft Meta Data

When building a business intelligence solution based on PeopleSoft data it can be useful to know the following:

* What is the non-technical description or business name for a particular column in a table?
* Does there exist a translate value (language-specific descriptive text for a given code) for a particular column in a table?

The query below (Meta data query) illustrates how the PeopleSoft system can be queried in order to find the relevant non-technical descriptions for the total set of columns in a table. The same query also performs an outer join to the translate table, PSXLATITEM, to see if any column contains codes which can be translated to a language specific text. An example of a translate value would be a MARITAL\_STATUS value of ”M.” We can select from the PSXLATITEM table to determine that a code of ”M” can be translated to the text ”Married.”

The query below produces output which contains a list of columns for the given table “PS\_DEPT\_TBL” in this example. This list also contains the non-technical description, the name of the column and a value of ‘Y’ if the column can be translated using the PSXLATITEM table. To execute this query for another table simply replace the string constant ”DEPT\_TBL” with the name of the appropriate table (removing the ”PS\_” prefix) ; e.g.“‘GL\_ACCOUNT\_TBL.”

This set of information can be used when constructing tables (using non-technical column names), Reports and Analysis Services cubes (for dimension attribute names, etc). It is also useful during the Extract, Transform, and Load (ETL) process where coded-column values can be translated into language-specific text using a lookup to the PSXLATITEM table.

**Meta data query**

SELECT PSRECDEFN.RECNAME, PSRECDEFN.RECDESCR, PSDBFIELD.FIELDNAME,

PSDBFIELD.DESCRLONG, DOMAINTAB.DOMAIN

FROM PSRECDEFN, PSRECFIELD, PSDBFIELD

LEFT OUTER JOIN (SELECT DISTINCT FIELDNAME, 'Y' AS DOMAIN FROM PSXLATITEM) AS DOMAINTAB

ON DOMAINTAB.FIELDNAME = PSDBFIELD.FIELDNAME

WHERE PSRECDEFN.RECNAME = 'DEPT\_TBL'

AND PSRECDEFN.RECNAME = PSRECFIELD.RECNAME

AND PSRECFIELD.FIELDNAME = PSDBFIELD.FIELDNAME

### Understanding Trees

Trees are a fundamental part of financial reporting. They enable data to be aggregated and filtered according to hierarchical structures created of entities such as accounts, departments, and products. When building a business intelligence solution for PeopleSoft, it is therefore important to understand firstly, how the tree tables work, and secondly, how to extract the trees for use in reporting and analysis outside of the PeopleSoft application.

PeopleSoft trees support unbalanced and ragged hierarchies. As such, they are stored in a particular way which is not all that helpful for reporting. For example it is fairly normal in reporting to say “show me all sales by product category.” In PeopleSoft the “product category” may be represented as a level in a tree. In order to aggregate all ”sales” data by ”product category” we would need to find all of the tree nodes at a given level and then join this to the sales data. This is complicated by the fact that the tree table which holds these nodes, PSTREENODE, is a recursive (or parent-child) table. It is therefore required that you perform a recursive join on this table to find all children and leaf level nodes (records) before joining to the sales table. This is not something that is easily achieved using SQL Server.

The most important tables to understand when dealing with trees are:

PSTREEDEFN: The master record for the tree.

PSTREENODE: Parent-child/recursive table that provides access to all of the nodes in a tree (all of the component parts of the tree branches).

PSTREELEAF: This table provides the set of leaf-level or detail records for a tree node. Not all nodes have leaves, indeed some trees may not contain any leaves at all. In addition, leaves can be attached to any node at any level of a tree.

The following sequence of queries provides a sample of how to access tree data across these tables.

#### Step 1: Identify the tree

**SELECT SETID, TREE\_NAME, EFFDT, EFF\_STATUS, TREE\_STRCT\_ID, DESCR, TREE\_HAS\_RANGES**FROM PSTREEDEFN  
WHERE SEID = 'SHARE'  
AND TREE\_NAME = 'ACCTROLLUP'  
AND EFFDT = '01/01/1900'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SETID | TREE\_NAME | EFFDT | EFF\_STATUS | TREE\_STRCT\_ID | DESCR | TREE\_HAS\_RANGES |
| SHARE | ACCTROLLUP | 1900-01-01 | A | ACCOUNT | Account Tree | Y |

The query above retrieves the basic tree definition from the PSTREEDEFN table. Important items returned by this query are ”TREE\_HAS\_RANGES” which determines whether a tree has leaves based on ranges of numbers, e.g. ranges of account codes, and ”TREE\_STRCT\_ID” which determines which other tables to use to find tree information, e.g. this query enables us to find the table containing the tree leaves (detail records).

SELECT TREE\_STRCT\_ID, DTL\_RECNAME  
FROM PSTREESTRCT  
WHERE TREE\_STRCT\_ID = 'ACCOUNT'

|  |  |
| --- | --- |
| TREE\_STRCT\_ID | DTL\_RECNAME |
| ACCOUNT | GL\_ACCOUNT\_TBL |

The value of ‘”REE\_STRCT\_ID” is then used to qualify the above query which returns the name of the detail table which is used to find the tree leaves. This is important in the next step. By prefixing ”PS\_”’ to the value returned as ”DTL\_RECNAME” it is possible to find the table which will be used to define the tree leaves.

#### Step 2: Determine whether the tree has leaves

In Step 1 the query returned a value “TREE\_HAS\_RANGES.” In the example, the value returned was ”Y” or Yes. This query determines that in order to find the set of leaves for a give tree node, you will be required to join to the detail or leaf table (PS\_GL\_ACCOUNT\_TBL in the example) using a range-based join. This is demonstrated in Step 3a.

#### Step 3a: Query a tree with leaves

The following query therefore uses the information gathered during Step 1 and Step 2 to retrieve a list of parent and child nodes (including all leaves) which can be used in a BI solution.

The query is divided into two parts. Part 1 retrieves the list of parent nodes and child nodes from the PSTREENODE table using the required SETID, TREE\_NAME, and EFFDT.

Using this example, it is known that this tree has ranges (PSTREEDEFN.TREE\_HAS\_RANGES = ”Y’”) and that the tree detail table is PS\_GL\_ACCOUNT\_TBL (PSTREESTRCT.DTL\_RECNAME = ”GL\_ACCOUNT\_TBL”). Part 2 therefore further retrieves the list of nodes and leaves, where the leaves are provided by performing a range-based join to the PS\_GL\_ACCOUNT\_TBL table. Note that as we are joining to the PS\_GL\_ACCOUNT\_TBL table, which is a control table, we will need to specify an appropriate SETID and EFFDT value. In this example we use the same SETID as the tree, and find the maximum EFFDT value less than or equal to the tree EFFDT. Alternatively, we could have used today’s date (GETDATE()) in order to use the current definition of the Account.

SELECT PSTREENODE.SETID, PSTREENODE.TREE\_NAME, PSTREENODE.EFFDT, RTRIM(TREE\_NODE) as TREENODE, RTRIM(PARENT\_NODE\_NAME) as PARENTNODE, RTRIM(TREE\_NODE) as NODENAME   
FROM PSTREENODE   
**WHERE** PSTREENODE.SETID = 'SHARE'   
AND RTRIM(TREE\_NAME) = 'ACCTROLLUP'   
AND EFFDT = '01/01/1900'  
UNION ALL  
SELECT A.SETID, N.TREE\_NAME, A.EFFDT,A.ACCOUNT AS TREENODE, RTRIM(ISNULL(N.TREE\_NODE,'N/A')) AS PARENTNODE, A.DESCR AS NODENAME   
FROM PSTREENODE N   
INNER JOIN PSTREELEAF L ON N.SETID = L.SETID   
AND N.TREE\_NAME = L.TREE\_NAME   
AND N.SETID = 'SHARE'   
AND N.TREE\_NAME = 'ACCTROLLUP'   
AND N.EFFDT = '01/01/1900'  
AND N.EFFDT = L.EFFDT  
AND N.TREE\_NODE\_NUM = L.TREE\_NODE\_NUM   
INNER JOIN PS\_GL\_ACCOUNT\_TBL A ON A.ACCOUNT >= L.RANGE\_FROM   
AND A.ACCOUNT <= L.RANGE\_TO   
AND A.SETID = L.SETID  
AND A.EFFDT = (

SELECT MAX(EFFDT)   
FROM PS\_GL\_ACCOUNT\_TBL M   
WHERE M.ACCOUNT = A.ACCOUNT   
AND M.SETID = L.SETID   
AND M.EFFDT <= '01/01/1900'

)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SETID | TREE\_NAME | EFFDT | TREENODE | PARENTNODE | NODENAME |
| SHARE | ACCTROLLUP | 1900-01-01 | ALLACCOUNTS |  | ALLACCOUNTS |
| SHARE | ACCTROLLUP | 1900-01-01 | BALSHEET | ALLACCOUNTS | BALSHEET |
| SHARE | ACCTROLLUP | 1900-01-01 | ASSETS | BALSHEET | ASSETS |
| SHARE | ACCTROLLUP | 1900-01-01 | CURASSETS | ASSETS | CURASSETS |
| … | … | … | … | … | … |

#### Step 3b: Query a tree with no leaves

The following example demonstrates a query which would be used if a tree has no leaves.

SELECT PSTREENODE.SETID, PSTREENODE.TREE\_NAME, PSTREENODE.EFFDT, RTRIM(TREE\_NODE) as TREENODE, RTRIM(PARENT\_NODE\_NAME) as PARENTNODE, RTRIM(TREE\_NODE) as NODENAME   
FROM PSTREENODE   
**WHERE** PSTREENODE.SETID = 'SHARE'   
AND RTRIM(TREE\_NAME) = 'ACCOUNTING PERIOD'   
AND EFFDT = '01/01/1900'  
UNION ALL  
SELECT A.SETID, N.TREE\_NAME, N.EFFDT,A.YEAR\_PERIOD\_STRING AS TREENODE, RTRIM(ISNULL(N.TREE\_NODE,'N/A')) AS PARENTNODE, A.DESCR AS NODENAME   
FROM PSTREENODE N   
INNER JOIN PSTREELEAF L ON N.SETID = L.SETID   
AND N.TREE\_NAME = L.TREE\_NAME   
AND N.SETID = 'SHARE'   
AND N.TREE\_NAME = 'ACCOUNTING PERIOD'   
AND N.EFFDT = '01/01/1900'  
AND N.EFFDT = L.EFFDT  
AND N.TREE\_NODE\_NUM = L.TREE\_NODE\_NUM   
INNER JOIN PS\_YEAR\_PERIOD\_TBL A ON A.YEAR\_PERIOD\_STRING >= L.RANGE\_FROM   
AND A.SETID = L.SETID

The queries 3a and 3b therefore provide a list of parent nodes and their child nodes for a given version (EFFDT) of a specific tree. We can then use this information in the construction of our data mart by flattening out this list (to provide a dimension table which has one column per hierarchy level) or by adding additional rows to aid aggregation (adding rows to translate the parent-child table to an ancestor-descendent table which contains the transitive closure between every node of the tree).

### Impact and Correct Use of SETIDs and Effective Dates

The two keys that organize the entire PeopleSoft database are SETID and BUSINESS\_UNIT. One or both of these can be found on almost all base and control tables in the database. Many organizations will implement only one value for BUSINESS\_UNIT in an organization; however, it can be used to implement multiple sets of books or multiple legal entities. Its use (of more than one default value) is optional. The SETID is the key value for what PeopleSoft calls “TableSets.” These are lists of values. Their use is to enable entities to be defined differently according to their use (e.g. Department ”001” may have a different definition according to SETID values ”SHARE” and ”USA01”).

Business units and SETIDs work together. When you are working with base tables in Financials (e.g. PS\_LEDGER) you will find the column BUSINESS\_UNIT. You will also find the column ACCOUNT. If you try to join the table PS\_LEDGER to the table PS\_GL\_ACCOUNT\_TBL (PS\_LEDGER.ACCOUNT = PS\_GL\_ACCOUNT\_TBL.ACCOUNT) you will immediately see a problem if you have a system with more than one BUSINESS\_UNIT value or more than one SETID value. That is, the key of the PS\_GL\_ACCOUNT\_TBL table consists of SETID, ACCOUNT, and EFFDT. This means that you must resolve two issues in order to join the PS\_LEDGER table to the PS\_GL\_ACCOUNT\_TBL table:

1. How to join using the correct SETID?
2. How to specify the correct EFFDT (Effective Date)?

In order to find the correct SETID value, the Set Control Record table is used, PS\_SET\_CNTRL\_REC. Using this table you can find the correct SETID for a given context for a given business unit. For example:

SELECT PS\_LEDGER.\*, PS\_GL\_ACCOUNT.\*

FROM PS\_SET\_CNTRL\_REC, PS\_LEDGER, PS\_GL\_ACCOUNT\_TBL

WHERE PS\_LEDGER.ACCOUNT = PS\_GL\_ACCOUNT\_TBL.ACCOUNT

AND **PS\_LEDGER.BUSINESS\_UNIT = PS\_SET\_CNTRL\_REC.SETCNTRLVALUE**

AND **PS\_SET\_CNTRL\_VALUE.SETID = PS\_GL\_ACCOUNT\_TBL.SETID**

AND **PS\_SET\_CNTRL\_VALUE.RECNAME = ‘GL\_ACCOUNT\_TBL’**

The query above (joins to the Set Control Record table are highlighted in bold) enables the PS\_LEDGER table to be joined to the PS\_GL\_ACCOUNT\_TBL table using the correct account SETID for the BUSINESS\_UNIT referenced on the Ledger table. However, there remains a problem with this query, namely that the table PS\_GL\_ACCOUNT\_TBL contains the column EFFDT as part of its primary key. This enables the same ACCOUNT definition to change over time, where the history is kept and a new record is created with a new effective date (EFFDT). A single account code can therefore consist of many rows in this table, each with a different EFFDT value. A common method of using the EFFDT field is to use the latest definition according to the current date (you may also wish to use the definition according to the posted date of the transaction). The query is therefore modified as follows:

SELECT PS\_LEDGER.\*, PS\_GL\_ACCOUNT.\*

FROM PS\_SET\_CNTRL\_REC, PS\_LEDGER, PS\_GL\_ACCOUNT\_TBL

WHERE PS\_LEDGER.ACCOUNT = PS\_GL\_ACCOUNT\_TBL.ACCOUNT

AND **PS\_LEDGER.BUSINESS\_UNIT = PS\_SET\_CNTRL\_REC.SETCNTRLVALUE**

AND **PS\_SET\_CNTRL\_VALUE.SETID = PS\_GL\_ACCOUNT\_TBL.SETID**

AND **PS\_SET\_CNTRL\_VALUE.RECNAME = ‘GL\_ACCOUNT\_TBL’**

AND **PS\_GL\_ACCOUNT\_TBL.EFFDT = (**

**SELECT MAX(EFFDT) FROM PS\_GL\_ACCOUNT\_TBL MX**

**WHERE MX.ACCOUNT = PS\_GL\_ACCOUNT\_TBL.ACCOUNT**

**AND MX.SETID = PS\_GL\_ACCOUNT\_TBL.SETID**

**AND MX.EFFDT <= GETDATE() )**

This concept is applied in many places. It is common to require a lookup to the Set Control Record table whenever you are joining a base table to a control table. It is also common that control tables contain effective dated data. This means that when accessing a table such as PS\_LEDGER and joining this to many control tables e.g. PS\_DEPT\_TBL, PS\_PROJECT\_TBL, PS\_GL\_ACCOUNT\_TBL, etc. it will be necessary to join to the Set Control Record table many times in order to find the correct SETID to use for each of these control tables (as it may not be a single SETID used across all tables).

As you see, incorrect use of SETIDs and EFFDTs can result in too much data (partial Cartesian product) or too little data (incorrect SETID used). It is important when extracting data to resolve these items correctly. One advantage of a BI solution which utilizes a data mart is that this resolution can take place once during ETL and thereafter leave a relatively simple join between surrogate keys.

## The PeopleSoft Financials Data Mart

The solution demonstrator provides an example PeopleSoft Financials Data Mart (multiple star schema). This is an example solution which can be used in place of part of the PeopleSoft EPM Performance Warehouse. However, this solution is not intended to replace the PeopleSoft EPM solution; it is a demonstrator. It is limited in scope. Organizations could consider using this solution demonstrator as the basis of developing their own Microsoft technology-based data warehouse or data mart for PeopleSoft. The demonstrator therefore delivers core capabilities such as de-normalization of data, surrogate key generation, translate table lookup, resolution of setID indirection, proper effective date handling, and tree extraction. These capabilities can be built upon and extended to meet the needs of an individual organization.

### Model Overview

The SQL Server 2005 data mart is delivered based on two levels:

1. A series of tables which mimic their source tables in PeopleSoft, with the addition of a sequentially-generated surrogate key (integer) for dimension tables.
2. A series of views which generate an ”effective to” date given the supplied ”effective from” date (EFFDT), adds a default row (key value of ”-1”’), performs any lookups to the translation-text table (PSXLATITEM) and for the fact tables, and resolves the correct set of surrogate keys in order to join to each dimension table by performing a number of dimension table lookups using the correct method of SETID indirection based on BUSINESS\_UNIT, including correct use of EFFDT.

The example data model provides coverage of the following subject areas:

* Billing (AR)
* Payments, Voucher, Cash, and Schedules (AP)
* GL, Journal, and Budget (GL)

These subject areas are delivered across nine fact tables with their associated dimension tables. Many dimension tables are shared across multiple fact tables.

Note that the schema is supplied with no additional indexes (other than primary key). You may be required to tune this setup according to use. Much of the ”real work” of the data model also takes place in the ”database view layer.” You may also be required to materialize these views if this schema is to be used to support relational reporting (as opposed to being used purely as a data source for the population of an Analysis Services database).

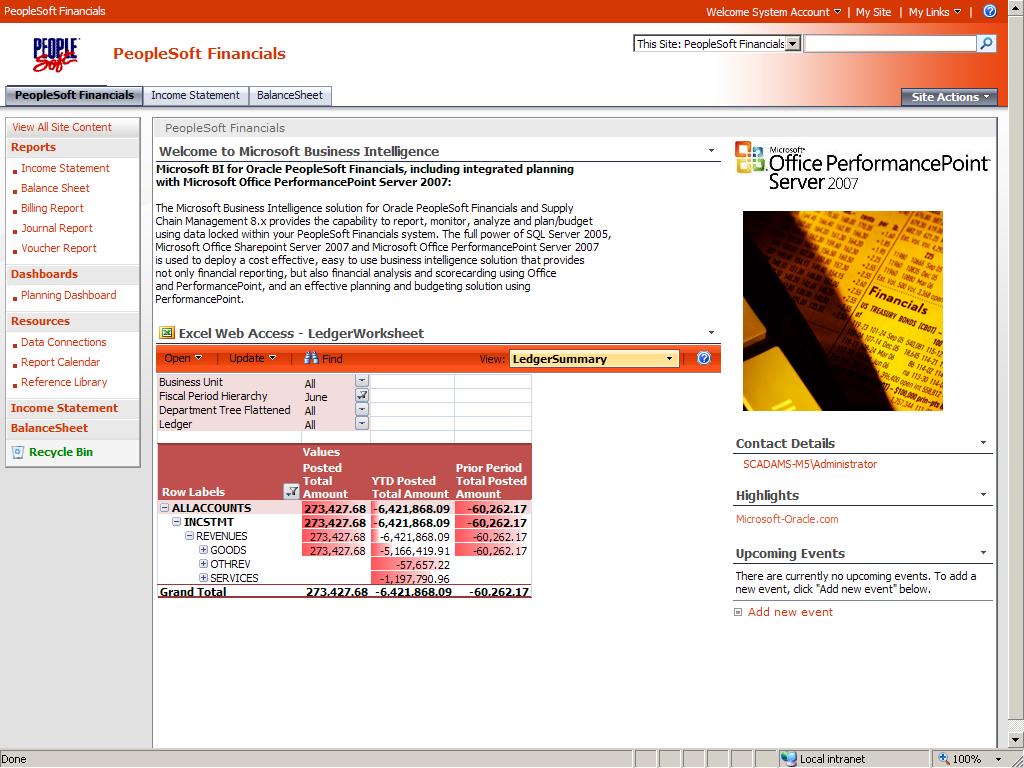
# PeopleSoft Enterprise 8.8 Financials Solution Demonstrator

## Installation

Note: The construction of this solution demonstrator is based on the Oracle PeopleSoft Enterprise 8.8 FIN/SCM demonstration database running on Microsoft SQL Server 2005. To use this demonstrator with an organization’s own systems, or a source database platform other than SQL Server, it may be necessary to make modifications to the Integration Services packages supplied. To install and use the solution demonstrator:

1. Extract the zip file **PeopleSoftEnterpriseR2.zip.**
2. Explore the unzipped contents.
3. Locate and extract the file **PeopleSoft.zip** which can be found in **<extract dir>\PeopleSoft Enterprise R1\Source\Backups.**
4. Restore the extracted database backup **PSFIN\_Source.bak** as database **”EP.”**
5. Execute the following scripts using SQL Server Management Studio:  
   Scripts can be found in **<extract dir>\PeopleSoft Enterprise R1\Source\DDL and Data:**
   1. **Create\_Database.sql**
   2. **Create\_Tables\_and\_Views.sql**
6. Open the BI project **<extract dir>\PeopleSoft Enterprise R1\VS Project Files\PeopleSoft Enterprise 8.x.sln.**
   1. Change any connection strings in all projects if you do not want to deploy to **localhost.**
   2. Run SQL Integration Services packages in the following order:
      1. Execute all **Dim \*.dtsx.**
      2. Execute all **Dim \* Tree.dtsx.**
      3. Execute all **Fact \*.dtsx.**
   3. Deploy and process Analysis Services project **”PeopleSoft.”**
   4. Deploy Reporting Services project (change server and port number from 88).
7. You may now execute the sample reports supplied using SQL Reporting Services Report Manager.
8. Using PerformancePoint Server 2007 Dashboard Designer, open the file **PeopleSoft Financials Dashboard.bswx** from location **<extract dir>\PeopleSoft Enterprise R1\PPS.** It will be necessary to first modify the **<ScorecardModel/>** element of the XML file to change the scorecard server URL if required (the default value is ”localhost”). The element to be changed is **ScorecardServerURL.**. You may also deploy the Dashboard **PeopleSoft Planning Dashboard.bswx** which can be found in the same location; however this dashboard has a dependency on Point 9 below, which is the deployment of the Planning application sample.
9. To use the PerformancePoint Server 2007 Planning application:
   1. Import the application database using the PPSCMD command line parameters. An example is shown below.
   2. PpsCmd.exe ImportXml /server [http://localhost:46787](http://www.tdwi.org/research/display.aspx) “<extract dir>\PeopleSoft Enterprise R2\PPS \Planning\PeoplSoft Financials.xml”. This will create the application and the application DB.
   3. Create the StagingDB for this application from the Admin Console.
   4. Restore the StagingDB supplied as <extract dir>\PeopleSoft Enterprise R2\PPS \Planning\PeoplSoft Financials\_StagingDB.bak (this will be copied over the blank StagingDB created above).
   5. Run the Planning Server Data Migration Wizard tool to load from the StagingDB. When you run the tool, there are two options - choose the option: Load from staging DB.
   6. You will have to manually create users, give them security rights, and create jobs and assignments. You may wish to create a cycle for 01 Jan 2001 to 31 Dec 2001 using the Scenario BUDGETS. This will match the form supplied “Formatted Expense Data Entry Form” which can be used for the form assignment for the cycle.

**Note**: Backup files are also supplied for the SQL Server data mart and the Analysis Services database should you wish to skip the steps of deploying and building these components of the solution.

The final solution can be deployed to Microsoft Office SharePoint Server 2007 using your favorite theme and styles. The screenshot below demonstrates a typical SharePoint site.

## Integration Services Packages

Included in the Microsoft Visual Studio® solution located at **<zip root>\VS Project Files\PeopleSoft Enterprise 8.x.sln** you will find a SQL Server 2005 Integration Services project. This project contains a number of Integration Services packages which can be used to extract data from PeopleSoft core tables and populate tables which form the PeopleSoft Financials data mart. The following section lists some important points of customization in these packages, and also provides the list of tables from which data will be extracted (the database login/user will therefore require SELECT access to these tables).

### Package Customization

The following section lists some important points of customization. These are areas which will require modification if the supplied packages are to be executed on any source other than the PeopleSoft Demonstration system supplied as part of the installation.

**Load Dim Calendar.dtsx**

**Load Dim Date** data flow: Change CALENDAR\_ID on data source task. The CALENDAR\_ID used should reflect that used in your organization and that which you wish to use for reporting of fiscal periods.

**Load Dim \* Tree.dtsx**

**Load \* Tree Tables** data flow: **Read PeopleSoft Tre**e data source task: adjust SQL for SETID, TREE\_NAME and EFFDT. Change these in the SQL Server script to reflect the tree (and version) you wish to use for reporting.

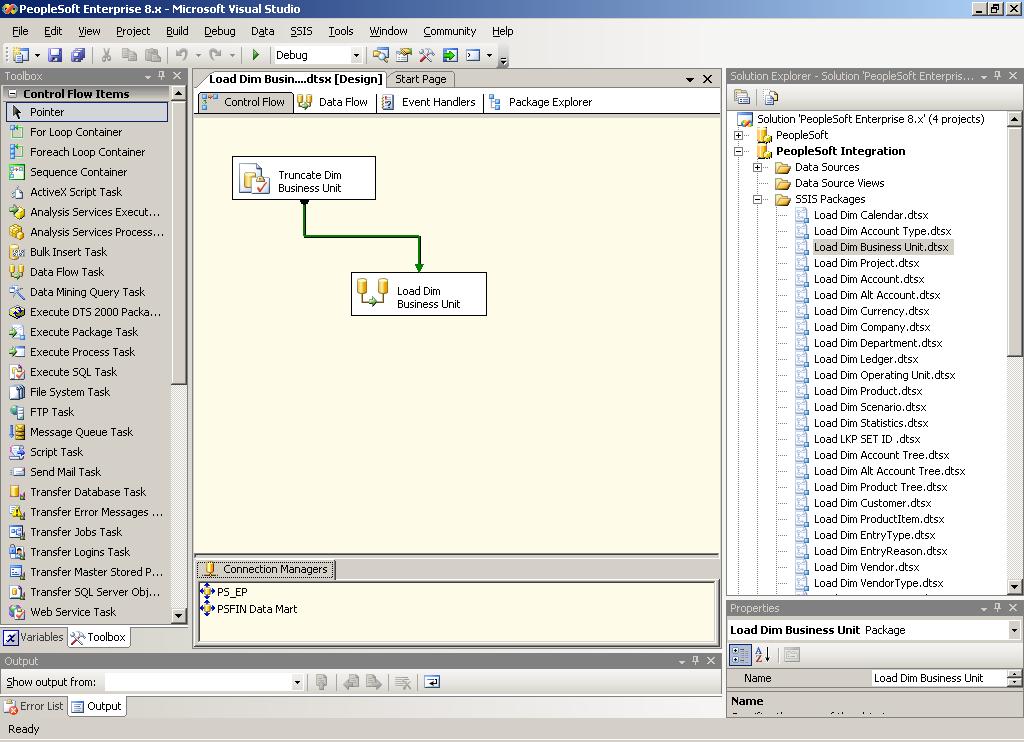
**Load \* Tree Tables** data flow: **Make Flattened Hierarchy** script task: if your tree is deeper than 10 levels on any branch you should extend this script. You will need to modify the code to add additional mappings to output variables. You will also be required to add outputs beyond 10 levels (levels 0-9 are included in the script).

In addition, for all tree scripts If you need more than one SETID, TREE\_NAME, and/or EFFDT (more than one tree or version for reporting on any of the dimensions, e.g. you need two different Account trees for reporting), add a loop into the package and use variables in place of the values coded into the SQL Server script.

### Package Execution

The Integration Services packages supplied in the Solution Demonstrator project must be executed following a specific order. The execution order is listed below. You may wish to create a single Integration Services package to run each of these packages in the specified order, however this “master execution package” is not supplied as part of the demonstrator.

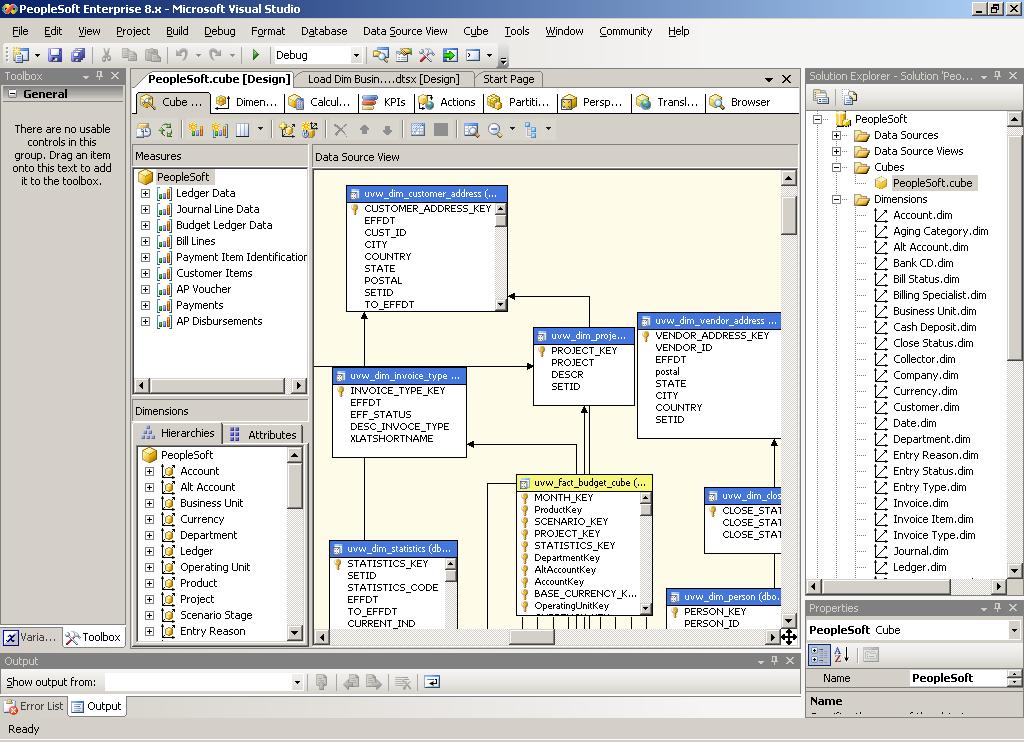
1. Execute all **Dim \*.dtsx.**
2. Execute all **Dim \* Tree.dtsx.**
3. Execute all **Fact \*.dtsx.**



### List of Source Tables

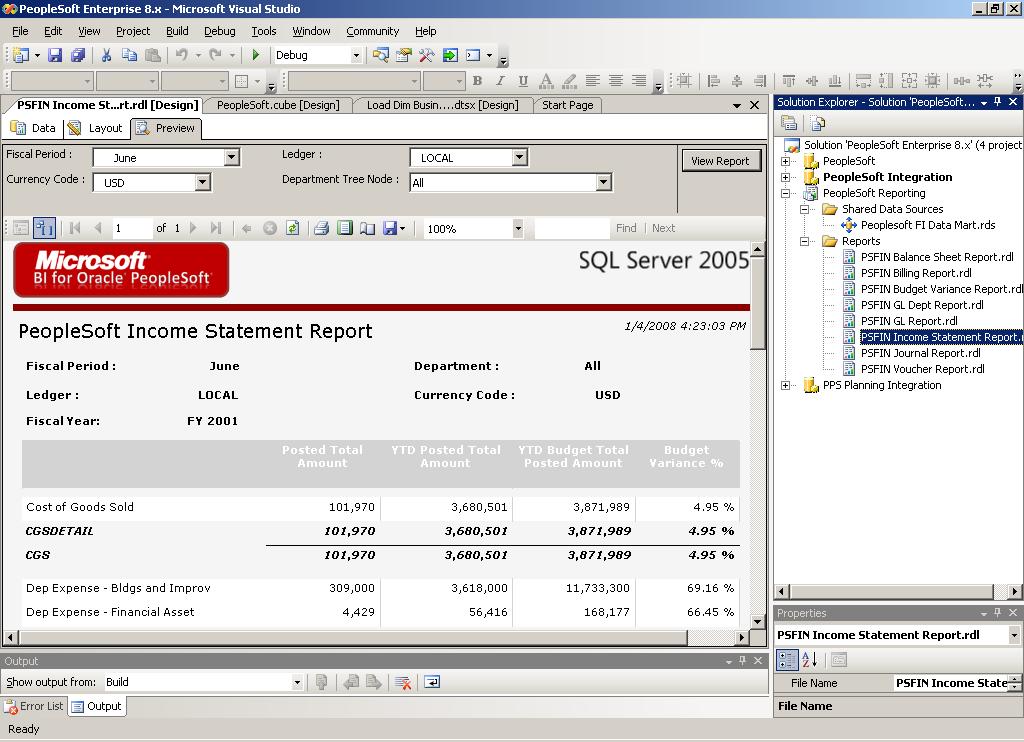
|  |
| --- |
| PEOPLESOFT 8.8 TABLE |
| PS\_CAL\_DETP\_TBL |
| PS\_ACCT\_TYPE\_TBL |
| PS\_BUS\_UNIT\_TBL\_GL |
| PS\_PROJECT\_ID\_VW |
| PS\_GL\_ACCOUNT\_TBL |
| PS\_ALTACCT\_TBL |
| PS\_CURRENCY\_CD\_TBL |
| PS\_COMPANY\_TBL |
| PS\_DEPT\_TBL |
| PS\_LED\_DEFN\_TBL |
| PS\_OPER\_UNIT\_TBL |
| PS\_PRODUCT\_TBL |
| PS\_BD\_SCENARIO\_TBL |
| PS\_STAT\_TBL |
| PS\_SET\_CNTRL\_REC |
| PSTREENODE |
| PSTREELEAF |
| PS\_CUSTOMER |
| PS\_PROD\_ITEM |
| PS\_ENTRY\_TYPE\_TBL |
| PS\_ENTRY\_REASN\_TBL |
| PS\_VENDOR |
| PS\_VENDOR\_TYPE |
| PS\_INV\_ITEMS |
| PS\_VENDOR\_ADDR |
| PS\_PAY\_TRMS\_TBL |
| PS\_PAY\_TRMS\_DSCNT |
| PS\_PAY\_TRMS\_NET |
| PS\_WTHD\_CD\_HDR |
| PS\_BANK\_CD\_TBL |
| PSXLATITEM |
| PS\_SPB\_PERSON\_TBL |
| PS\_AGING\_CATEG\_TBL |
| PS\_COLLECTOR\_TBL |
| PS\_BI\_SPECIALIST |
| PS\_CUST\_ADDRESS |
| PS\_CUST\_CONTACT |
| PS\_CUST\_CREDIT |
| PS\_VOUCHER |
| PS\_DISTRIB\_LINE |
| PS\_LEDGER |
| PS\_JRNL\_HEADER |
| PS\_JRNL\_LN |
| PS\_LEDGER\_BUDG |
| PS\_BI\_HDR |
| PS\_BI\_LINE |
| PS\_PAYMENT |
| PS\_PAYMENT\_ID\_ITEM |
| PS\_ITEM |
| PS\_ITEM\_ACTIVITY |
| PS\_PAYMENT\_TBL |
| PS\_PAYMENT |

## Analysis Services Cube

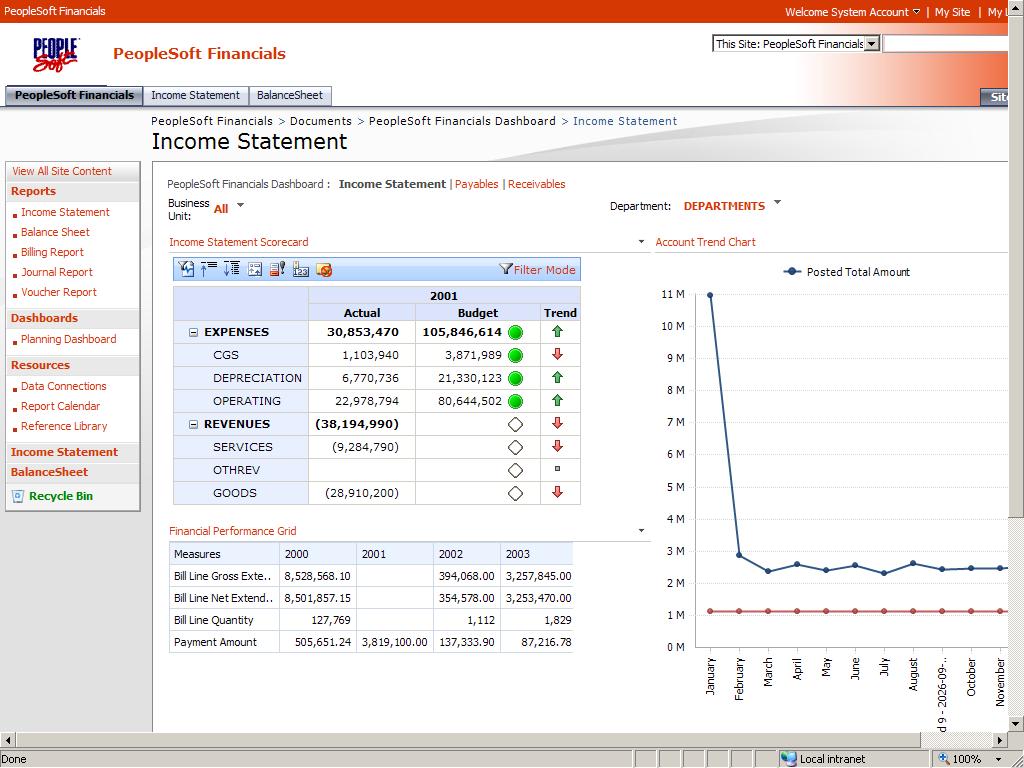
An example Analysis Services database/project is provided. The project is included in the solution **<extract dir>\PeopleSoft Enterprise R1\VS Project Files\PeopleSoft Enterprise 8.x.sln.**

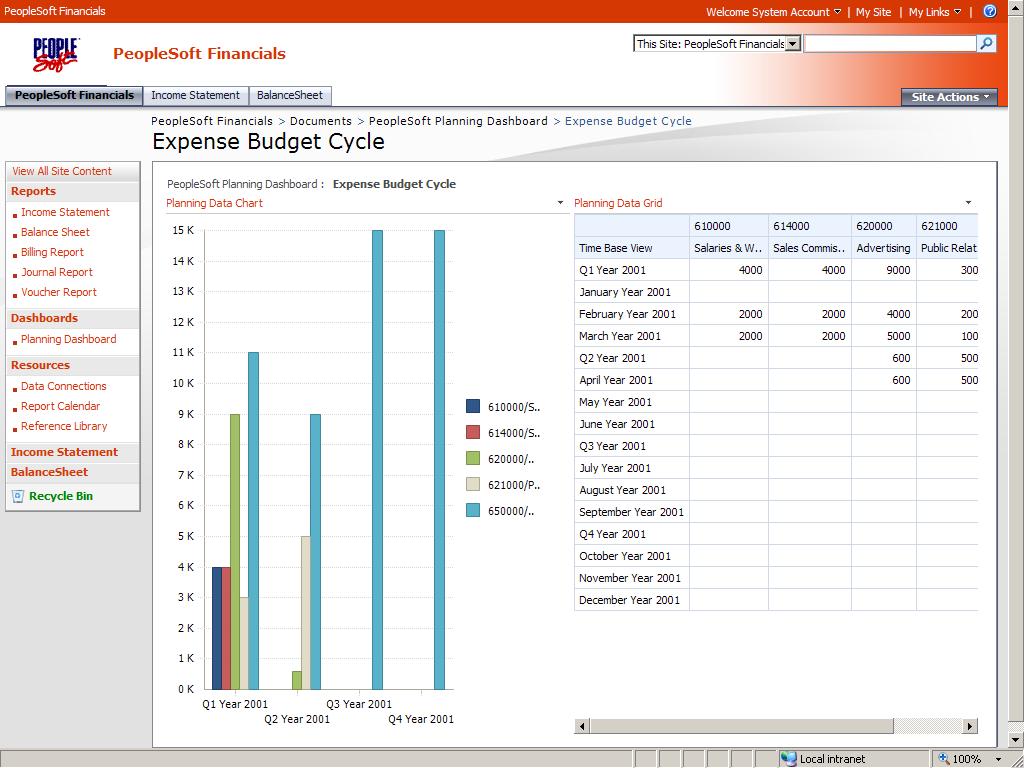
## Reporting Services Reports

A number of example Reporting Services reports are included in the solution **<extract dir>\PeopleSoft Enterprise R1\VS Project Files\PeopleSoft Enterprise 8.x.sln.** These reports are supplied based on the sample data provided by Oracle for PeopleSoft Financials/SCM. The reports therefore do not always contain parameters which would make sense for an organization (such as prompting for Business Unit or Operational Unit). These reports are therefore provided as samples only and should be customized if they are to be used as the basis for development of a solution.



## PerformancePoint Server Dashboards

Located in the directory **<extract dir>\PeopleSoft Enterprise R1\PPS** there is a sample PerformancePoint Dashboard. This dashboard uses only standard features of Office PerformancePoint Server and has no dependencies on other components such as ProClarity Analytic Server or Excel Services. This is done to ease deployment and sharing of this solution. It will therefore be desirable to add content to the dashboards which demonstrate other features of the Microsoft ”all–up”’ BI solution; including Excel Services, ProClarity, Reporting Services, and Microsoft Office Visio®.

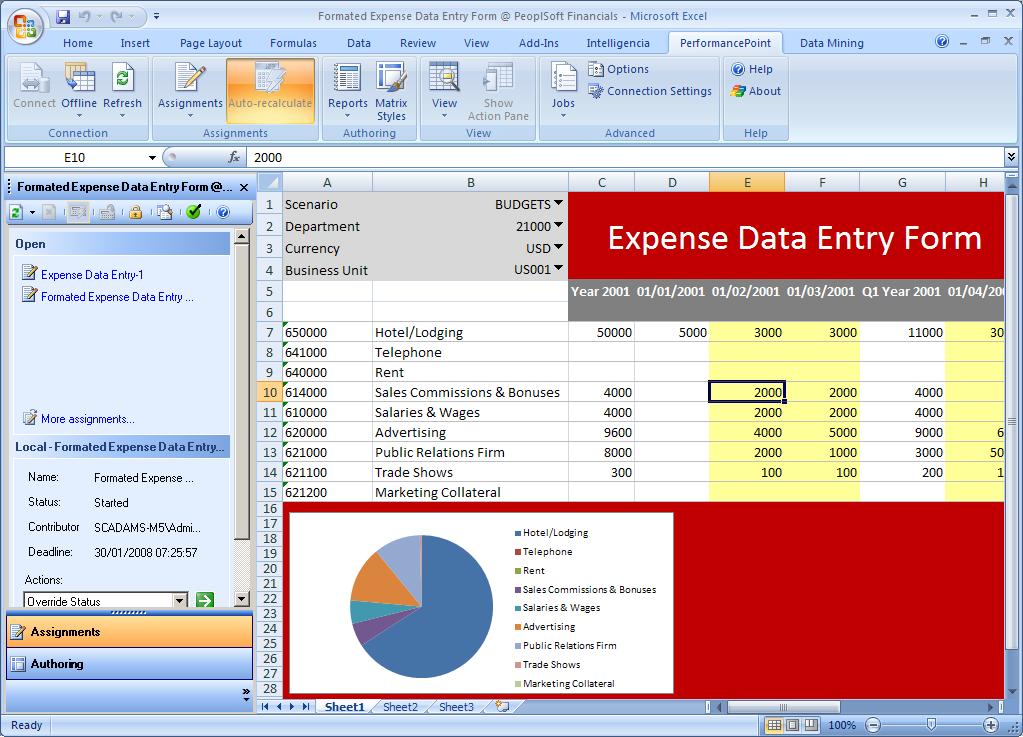
In addition to the above dashboard, an additional dashboard is supplied, **PeopleSoft Planning Dashboard.bswx**, that presents data (using a single Dashboard page) from the Planning application cube.

This enables updates to the “expense budget” scenario to be reflected on the dashboard.

## PerformancePoint Server Planning Application

Located in the directory **<extract dir>\PeopleSoft Enterprise R1\PPS**, the planning application supplied is based on a simple application that allows data entry of budget and forecast data (scenario) data, where the system is pre-populated with data from the Ledger and Budget Ledger tables in PeopleSoft. In addition, core dimensions such as Business Unit, Department, Account, Currency, and Scenario are built and populated using data from the PeopleSoft Financials data mart. The Department and Account dimensions include Member Sets which define the respective PeopleSoft Trees which have been extracted for these dimensions.

This application can be used to demonstrate a simple example, such as departmental expense budgeting. The form supplied illustrates this example. The user can enter a new budget for each of the months of 2001 for specific expense lines (accounts) for a specific business unit, currency, and department.



In addition, the Visual Studio solution <extract dir>\PeopleSoft Enterprise R2\VS Project Files\PeopleSoft Enterprise 8.x.sln contains an integration services project PPS Planning, which includes packages which load the Measure Group staging table from the Financials data mart, and which load the Account and Department hierarchy (tree) staging tables.

# Summary and Resources

Using this solution will help you to articulate the value of and to sell Business Intelligence solutions for Oracle PeopleSoft Enterprise applications. Below are additional links which will be of use when preparing to use this solution in a sales opportunity.

**Link to PeopleSoft and SQL Server 2005 BI solution demonstrator download:**

<http://www.microsoft-oracle.com/Assets/pdf/PeopleSoftEnterpriseR2.zip>

**Office PerformancePoint Server Resources:**

**AMR Research report: “Market Outlook: Microsoft’s Looming Impact on the Business Intelligence and Performance Management Market” dated 06/27/07**

<http://download.microsoft.com/download/6/0/B/60BBDFD1-D13E-4DE8-9D4F-629C122531F7/AMRresearch_MarketOutlook_impactofMSFTonBIandPM.pdf>

**Microsoft Web site: PerformancePoint Server 2007 Partner Resources**

<https://partner.microsoft.com/global/40050218?PS=95000124>

**Microsoft Web site: “Microsoft Business Intelligence and PerformancePoint Server”**

<http://www.microsoft.com/bi/products/performancepoint-overview.aspx>

**Office PerformancePoint Server Resources (contd.):**

**Microsoft Web site: “PerformancePoint Server Top Ten Benefits”**

<http://www.microsoft.com/business/performancepoint/productinfo/top10benefits.aspx>

**Microsoft Web site: “PerformancePoint Server Features and Functionality”**

<http://www.microsoft.com/business/performancepoint/productinfo/features.aspx>

**Microsoft Web site: “PerformancePoint Server Case Studies”**

<http://www.microsoft.com/business/performancepoint/productinfo/casestudies.aspx>

**PerformancePoint Server 2007 Planning Data Sheet**

<https://partner.microsoft.com/global/40050289>

**PerformancePoint Server 2007 Monitoring Data Sheet**

<https://partner.microsoft.com/global/40050291>

**PerformancePoint Server 2007 Analytics Data Sheet**

<https://partner.microsoft.com/global/40050290>

**PerformancePoint Server 2007 Management Reporting Data Sheet**

<https://partner.microsoft.com/global/40050292>

**PerformancePoint Server 2007 Data Sheet**

<http://download.microsoft.com/download/4/5/2/452d8197-1940-4442-8134-b6b82f57cfe8/PerformancePoint%20Server%20datasheet-Overview_10_07.docx>

**TechNet Webcast: Customer and Product Monitoring with PerformancePoint Server 2007**

<http://www.microsoft.com/events/EventDetails.aspx?CMTYSvcSource=MSCOMMedia&Params=%7eCMTYDataSvcParams%5e%7earg+Name%3d%22ID%22+Value%3d%221032360208%22%2f%5e%7earg+Name%3d%22ProviderID%22+Value%3d%22A6B43178-497C-4225-BA42-DF595171F04C%22%2f%5e%7earg+Name%3d%22lang%22+Value%3d%22en%22%2f%5e%7earg+Name%3d%22cr%22+Value%3d%22US%22%2f%5e%7esParams%5e%7e%2fsParams%5e%7e%2fCMTYDataSvcParams%5e>

**TechNet Webcast: PerformancePoint Server 2007 Technical Overview (Level 300)**

<http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?culture=en-US&EventID=1032355606&CountryCode=US>

**TechNet Webcast: Business Intelligence with Office PerformancePoint Server 2007 (Level 300)**

<http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?EventID=1032326209&EventCategory=5&culture=en-US&CountryCode=US>

**Office PerformancePoint Server Resources (cont’d.):**

**TechNet Webcast: Performance Management with PerformancePoint Server 2007 (Level 200)**

<http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?culture=en-US&EventID=1032353652&CountryCode=US>

**TechNet Webcast: Performance Management 101 with PerformancePoint Server 2007 (Level 200)**

<http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?EventID=1032323902&EventCategory=5&culture=en-US&CountryCode=US>

**ISV Innovation Briefing Webcast - Business Intelligence and PerformancePoint Server 2007 (Recording)**

<http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?EventID=1032349594&EventCategory=5&culture=en-CA&CountryCode=CA>

# Appendix

The appendix contains a source code listing (Microsoft Visual Basic® Script) for the Vertical and Horizontal or Flat tree scripts used as part of the Dim \*Tree.dtsx Integration Services packages. These scripts enable the PeopleSoft Tree structures to be flattened for use in reporting and analysis (resolving the list of parent-child relationships into a flat table with each tree level on a column, and a separate table which contains the transitive closure between each node of the tree i.e. an ancestor – descendent relationship).

**Vertical Tree Script**

' Microsoft SQL Server Integration Services user script component

' This is your new script component in Microsoft Visual Basic .NET

' ScriptMain is the entrypoint class for script components

Imports System

Imports System.Data

Imports System.Math

Imports Microsoft.SqlServer.Dts.Pipeline.Wrapper

Imports Microsoft.SqlServer.Dts.Runtime.Wrapper

Public Structure Node

Public setId As String

Public treeName As String

Public effDt As Date

Public parent As String

Public item As String

Public itemName As String

End Structure

Public Class LinkedList

Public oNode As Node

Public nextNode As LinkedList

Public Sub New(ByVal cNode As Node, ByVal pNode As LinkedList)

oNode = cNode

nextNode = pNode

End Sub

End Class

Public Class ScriptMain

Inherits UserComponent

Dim nodeList(5000) As LinkedList

Dim counter As Integer

Dim tempcounter As Integer

Public Overrides Sub Input\_ProcessInput(ByVal Buffer As InputBuffer)

Dim tempList As LinkedList

Dim tempPList As LinkedList

Dim tempPAList As LinkedList

Dim depth As Integer

tempcounter = 0

depth = 0

While Buffer.NextRow()

Input\_ProcessInputRow(Buffer)

End While

If Buffer.EndOfRowset Then

'loop the array and navigate the list

For tempcounter = 0 To (counter - 1)

depth = 0

tempList = nodeList(tempcounter)

With tempList.oNode

If .item > "" Then

'walk up each LinkedList outputing a row on each traversal

OutputBuffer.AddRow()

OutputBuffer.TREENAME = .treeName

OutputBuffer.SETID = .setId

OutputBuffer.EFFDT = .effDt

OutputBuffer.ANCESTOR = .item

OutputBuffer.ANCESTORNAME = .itemName

OutputBuffer.DESCENDENT = .item

OutputBuffer.DESCENDENTNAME = .itemName

OutputBuffer.DEPTH = 0

tempPList = tempList.nextNode

If Not IsNothing(tempPList) Then

OutputBuffer.ANCESTORPARENT = tempPList.oNode.item

OutputBuffer.ANCESTORPARENTNAME = tempPList.oNode.itemName

tempPAList = tempPList.nextNode

Else

OutputBuffer.ANCESTORPARENT = "TREE ROOT"

OutputBuffer.ANCESTORPARENTNAME = "Account Tree Root Node"

End If

While Not IsNothing(tempPList)

'output another row

depth = depth + 1

OutputBuffer.AddRow()

OutputBuffer.TREENAME = .treeName

OutputBuffer.SETID = .setId

OutputBuffer.EFFDT = .effDt

OutputBuffer.ANCESTOR = tempPList.oNode.item

OutputBuffer.ANCESTORNAME = tempPList.oNode.itemName

OutputBuffer.DESCENDENT = .item

OutputBuffer.DESCENDENTNAME = .itemName

OutputBuffer.DEPTH = depth

If Not IsNothing(tempPAList) Then

OutputBuffer.ANCESTORPARENT = tempPAList.oNode.item

OutputBuffer.ANCESTORPARENTNAME = tempPAList.oNode.itemName

tempPAList = tempPAList.nextNode

Else

OutputBuffer.ANCESTORPARENT = "TREE ROOT"

OutputBuffer.ANCESTORPARENTNAME = "Account Tree Root Node"

End If

tempPList = tempPList.nextNode

If Not IsNothing(tempPList) Then

tempPAList = tempPList.nextNode

End If

End While

End If

End With

Next

OutputBuffer.SetEndOfRowset()

End If

End Sub

Public Overrides Sub Input\_ProcessInputRow(ByVal Row As InputBuffer)

With Row

Dim tNode As New Node

Dim t2Node As Node

Dim found As Boolean = False

Dim foundItem As Integer = 0

Dim finder As Integer

tNode.setId = .SETID

tNode.treeName = .TREENAME

tNode.effDt = .EFFDT

tNode.item = .TREENODE

tNode.parent = .PARENTNODE

tNode.itemName = .TREENODENAME

'find

For finder = 0 To counter - 1

If nodeList(finder).oNode.item = .PARENTNODE Then

found = True

foundItem = finder

End If

Next

If found = True Then

Dim tList As New LinkedList(tNode, nodeList(foundItem))

nodeList(counter) = tList

Else

Dim tList As New LinkedList(tNode, Nothing)

nodeList(counter) = tList

End If

End With

counter = counter + 1

End Sub

End Class

**Flat Tree Script**

' Microsoft SQL Server Integration Services user script component

' This is your new script component in Microsoft Visual Basic .NET

' ScriptMain is the entrypoint class for script components

Imports System

Imports System.Data

Imports System.Math

Imports Microsoft.SqlServer.Dts.Pipeline.Wrapper

Imports Microsoft.SqlServer.Dts.Runtime.Wrapper

Public Structure Node

Public setId As String

Public treeName As String

Public effDt As Date

Public parent As String

Public item As String

Public itemName As String

End Structure

Public Class LinkedList

Public oNode As Node

Public parentArray(10) As Node

Public numParents As Integer

Public nextNode As LinkedList

Public Sub New(ByVal cNode As Node, ByVal pNode As LinkedList)

oNode = cNode

nextNode = pNode

End Sub

End Class

Public Class ScriptMain

Inherits UserComponent

Dim nodeList(5000) As LinkedList

Dim counter As Integer

Dim tempcounter As Integer

Public Overrides Sub Input\_ProcessInput(ByVal Buffer As InputBuffer)

Dim tempList As LinkedList

Dim tempPList As LinkedList

Dim tempPAList As LinkedList

Dim depth As Integer

Dim position As Integer

Dim poscounter As Integer

Dim level As Integer

tempcounter = 0

depth = 0

position = 0

While Buffer.NextRow()

Input\_ProcessInputRow(Buffer)

End While

If Buffer.EndOfRowset Then

For tempcounter = 0 To (counter - 1)

tempList = nodeList(tempcounter)

position = 0

tempList.parentArray(position) = tempList.oNode

tempPList = tempList.nextNode

While Not IsNothing(tempPList)

position = position + 1

tempList.parentArray(position) = tempPList.oNode

tempPList = tempPList.nextNode

End While

tempList.numParents = position + 1

Next

End If

If Buffer.EndOfRowset Then

'loop the array and navigate the list

For tempcounter = 0 To (counter - 1)

tempList = nodeList(tempcounter)

With tempList.oNode

If .item > "" Then

'walk up each LinkedList outputing a row on each complete walk up the tree

OutputBuffer.AddRow()

OutputBuffer.SETID = .setId

OutputBuffer.TREENAME = .treeName

OutputBuffer.EFFDT = .effDt

OutputBuffer.TREENODE = .item

OutputBuffer.NODENAME = .itemName

For poscounter = 0 To tempList.numParents - 1

level = ((poscounter - tempList.numParents) + 1) \* -1

If Not IsNothing(tempList.parentArray(poscounter)) Then

If level = 0 Then

OutputBuffer.L0TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L0NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 1 Then

OutputBuffer.L1TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L1NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 2 Then

OutputBuffer.L2TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L2NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 3 Then

OutputBuffer.L3TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L3NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 4 Then

OutputBuffer.L4TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L4NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 5 Then

OutputBuffer.L5TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L5NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 6 Then

OutputBuffer.L6TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L6NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 7 Then

OutputBuffer.L7TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L7NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 8 Then

OutputBuffer.L8TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L8NODENAME = tempList.parentArray(poscounter).itemName

ElseIf level = 9 Then

OutputBuffer.L9TREENODE = tempList.parentArray(poscounter).item

OutputBuffer.L9NODENAME = tempList.parentArray(poscounter).itemName

End If

End If

Next

End If

End With

Next

OutputBuffer.SetEndOfRowset()

End If

End Sub

Public Overrides Sub Input\_ProcessInputRow(ByVal Row As InputBuffer)

With Row

Dim tNode As New Node

Dim t2Node As Node

Dim found As Boolean = False

Dim foundItem As Integer = 0

Dim finder As Integer

tNode.setId = .SETID

tNode.treeName = .TREENAME

tNode.effDt = .EFFDT

tNode.item = .TREENODE

tNode.parent = .PARENTNODE

tNode.itemName = .NODENAME

'find

For finder = 0 To counter - 1

If nodeList(finder).oNode.item = .PARENTNODE Then

found = True

foundItem = finder

End If

Next

If found = True Then

Dim tList As New LinkedList(tNode, nodeList(foundItem))

nodeList(counter) = tList

Else

Dim tList As New LinkedList(tNode, Nothing)

nodeList(counter) = tList

End If

End With

counter = counter + 1

End Sub

End Class