User Guide **PUBLIC**

SAP PowerDesigner
Document Version: 16.6.6 – 2018-03-23

Enterprise Architecture

SAP PowerDesigner Documentation Collection

Content

1 Planning Your Enterprise Architecture Initiative

An enterprise architecture model (EAM) helps you analyze and document your organization, its capabilities and processes, the applications and systems that support them, and the physical architecture on which they are implemented.

Context

SAP® PowerDesigner® provides a robust solution for collecting, analyzing, communicating, and maintaining enterprise architecture metadata in order to:

- Align IT strategy with business goals, capabilities, and processes.
- Comply with legal standards or regulations requiring documentation of your organization's architecture.
- Document and communicate change management for:
  - Rationalization – seeking cost savings and efficiencies by standardizing and simplifying architectures.
  - Technology changes – where your system will be upgraded.
  - An acquisition or merger – where two disparate systems must be made to work together.

Note

You can create and edit enterprise architecture diagrams in your Web browser using PowerDesigner Web:

View, Create, and Edit	View Only

• Process maps
• Enterprise architecture diagrams - Unify the following types of diagram and support the creation of any enterprise architecture object:
  ○ Business communication diagrams
  ○ City planning diagrams
  ○ Application architecture diagrams
  ○ Service-oriented diagrams
  ○ Technology infrastructure diagrams
	• Organization charts
	• Timeline diagrams

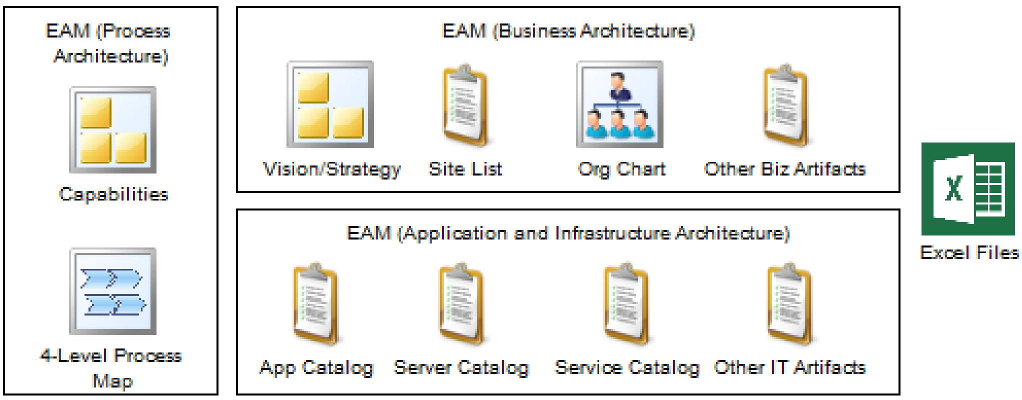
For detailed information about Web support for enterprise architecture, see *PowerDesigner Web > Enterprise Architecture*.

Procedure

1. Prepare your PowerDesigner landscape and repository to receive your metadata (see Preparing Your PowerDesigner Landscape and Repository [page 10]).
2. Decide what metadata you want to collect and how you want to structure it. The four broad domains of EA metadata are:
   - Organization Architecture - The people, sites, and structures of your organization (see Organization Architecture Modeling [page 36])
   - Business Architecture - What your organization does and how it does it (see Business Architecture Modeling [page 52])
   - Application Architecture - The software that your organization uses (see Application Architecture Modeling [page 66])
   - Infrastructure Architecture - The hardware and physical assets that your organization uses (see Infrastructure Architecture Modeling [page 87])

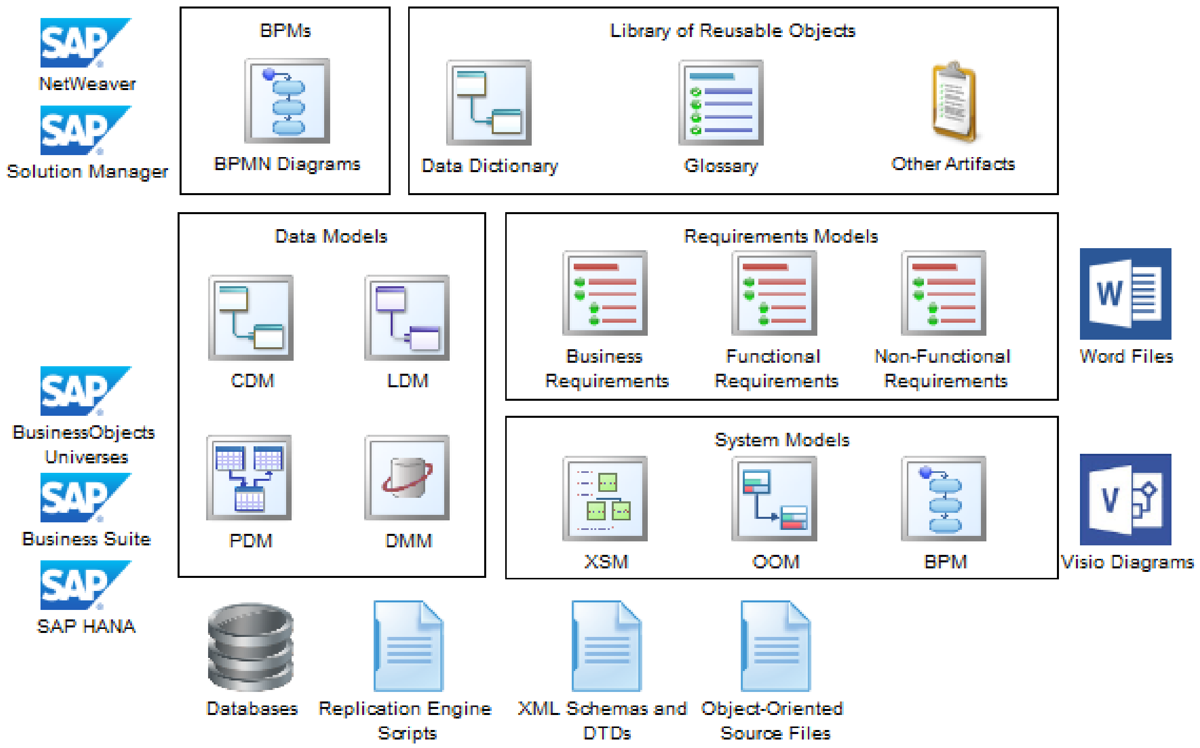
   > **Note**
   >
   > The `EA Example.eam` uses a subset of the available EA objects from each of these domains, structured in a particular way, and can be used as the starting point for your deliberations (see The EA Example Model, Excel Imports, and Charts [page 24]).

3. Create individual models (see Creating an EAM [page 11]) or an enterprise architecture framework project that combines your EAM diagrams with those of a Physical Data Model and other PowerDesigner modules (Creating an Enterprise Architecture Framework Project [page 14]) to begin modeling.
4. Import previously collated metadata where possible (see Importing EA Metadata from Excel [page 140] and Importing Visio Diagrams into PowerDesigner [page 141]).

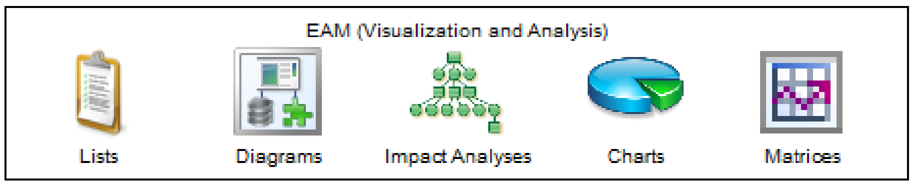


5. Connect your high-level EA models to your underlying landscape by inviting stakeholders, DBAs, developers, and implementers to create more detailed models of processes, databases, web services, and other assets.

   Business users who are process owners can document their processes in BPMN 2.0 business process diagrams using the Web browser-based PowerDesigner Web client (see *PowerDesigner Web*).



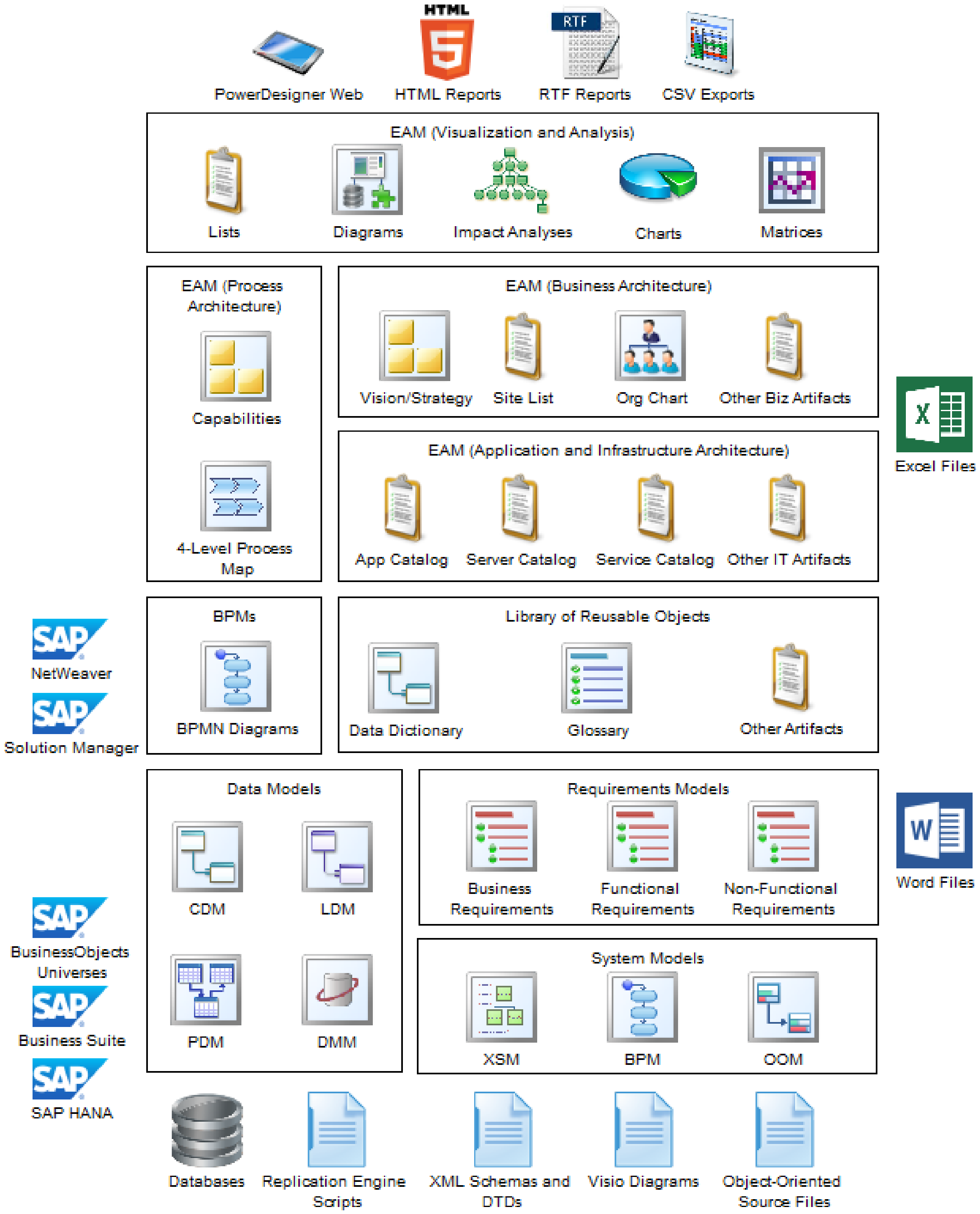
6. Create connections between your enterprise architecture and implementation models using:
    - Import and export wizards (see Exporting and Importing Objects to and from Other Models [page 143]).
    - Traceability links (see Traceability Links [page 147]).
    - Related diagrams (see Specifying Diagrams as Related Diagrams [page 149]).
7. Create appropriate viewpoints into your EA assets using:
    - Diagrams - Visualize and organize selected assets according to a particular need (see Creating EAM Diagrams [page 16] and Displaying EAM Objects in Diagrams [page 16]).
    - Charts - Analyze your assets based on datasets you define (see EA Example Charts [page 30]).
    - Impact Analyses - Show how changes to one or more objects will impact others to which they are linked (see *Core Features Guide* > *Linking and Synchronizing Models* > *Impact and Lineage Analysis*).
    - Dependency Matrices - Show connections between different types of objects (see *Core Features Guide* > *Modeling with PowerDesigner* > *Diagrams, Matrices, and Symbols* > *Dependency Matrices*).



8. Add information about how your corporate goals can be fulfilled through changes to your landscape, and the impacts that such changes will have on your assets (see Goal and EA Project Modeling [page 99]).
9. Communicate your EA metadata to stakeholders and the wider organization, using:

- PowerDesigner Web - Provides access to all your diagrams and metadata via a Web browser (see *PowerDesigner Web*).
- HTML and RTF Reports - Provide snapshots of your metadata suitable for documentation or inclusion in other reports (see *Core Features Guide > Modeling with PowerDesigner > Reports*).
- CSV Exports - For consumption by other reports or processes. Available from any list dialog.

10. Maintain your metadata in your repository and use it for ongoing planning, communication, and development:



1.1 Preparing Your PowerDesigner Landscape and Repository

The PowerDesigner repository provides a central point of storage for all your modeling assets, and can serve as the single source of truth for your enterprise metadata.

Procedure

1. Deploy your PowerDesigner repository to an appropriate server (see *Installation Guide > Planning Your PowerDesigner® Installation*) and connect to it.
2. Design an appropriate top-level folder structure for your repository to organize the models and other documents that will be stored there. For example, you may create the following folders:
    - Library - Pushes reference information and extensions and customization to all users (see *Core Features Guide > Modeling with PowerDesigner > The Browser > The Library* . Lists and catalogs of enterprise artifacts are commonly saved to the Library to promote their reuse in other models:
        - Glossary - Used for standardizing object names (see *Core Features Guide > Modeling with PowerDesigner > The Browser > The Glossary*).
        - Data Dictionary (CDM) - Defines common data elements (see *Data Architecture > Building Data Models > Conceptual and Logical Diagrams > Conceptual Diagrams > Example: Building a Data Dictionary in a CDM*).
        - Organization Architecture (EAM) - Sites, Organization Units, and People (see Organization Architecture Modeling [page 36]).
        - Resource Files - PowerDesigner resource files are often distributed through the library to ensure that all modelers are using the latest extensions and customizations (see Extending your Modeling Environment [page 34]).
    - EA Content - Can contain one or more high-level EA models and project documents, and may be branched to show the current or "as-is" situation and the projected "to-be" landscape.
    - EA Projects - Can contain one or more EA models drawing on objects in your landscape to show the goals that a project will contribute to fulfilling, and the assets that it will impact while doing so (see Goal and EA Project Modeling [page 99]).
    - Processes - Commonly contains an EAM model containing your 4-level process map and a folder structure to organize the business process models used to model in detail your level 4 processes (see Process Maps [page 52]).
    - Implementation Models - Can contain data models and other lower-level implementation models.
3. Create user accounts and set appropriate permissions based on who will need to consult, contribute to, or manage the metadata in each of your folders (see *Core Features Guide > Administering PowerDesigner > Controlling Repository Access*).

1.2 Creating an EAM

You create a new enterprise architecture model by selecting *File* > *New Model*.

Context

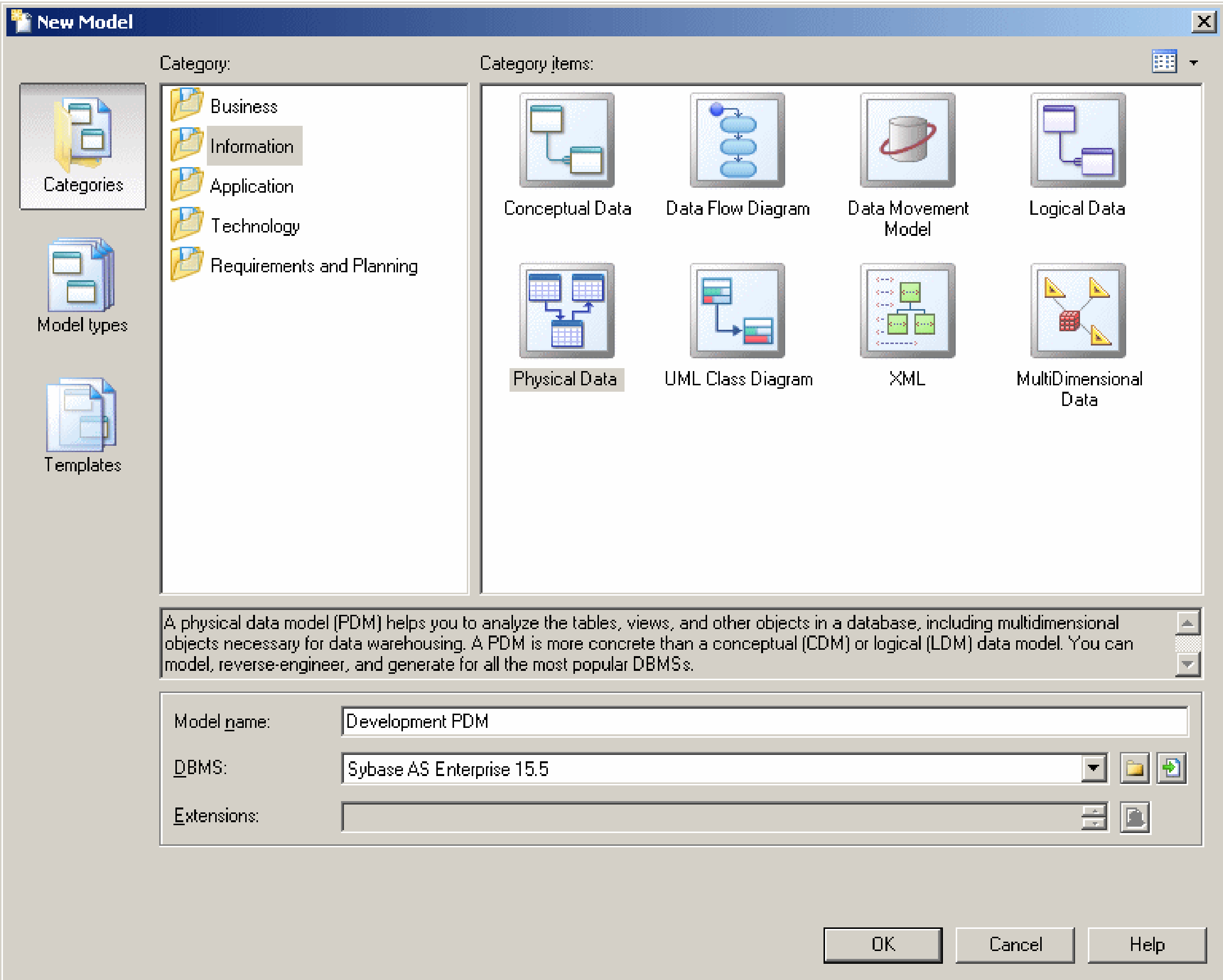
Note

In addition to creating an EAM from scratch with the following procedure, you can also create a model by importing:

- An Excel file (see Importing EA Metadata from Excel [page 140]).
- A Visio file (see Importing Visio Diagrams into PowerDesigner [page 141]).

The New Model dialog is highly configurable, and your administrator may hide options that are not relevant for your work or provide templates or predefined models to guide you through model creation. When you open the dialog, one or more of the following buttons will be available on the left hand side:

- *Categories* - which provides a set of predefined models and diagrams sorted in a configurable category structure.
- *Model types* - which provides the classic list of PowerDesigner model types and diagrams.
- *Template files* - which provides a set of model templates sorted by model type.



Procedure

1. Select *File* > *New Model* to open the New Model dialog.
2. Click a button, and then select a category or model type (*Enterprise Architecture Model*) in the left-hand pane.
3. Select an item in the right-hand pane. Depending on how your New Model dialog is configured, these items may be first diagrams or templates on which to base the creation of your model.

   Use the *Views* tool on the upper right hand side of the dialog to control the display of the items.
4. Enter a model name. The code of the model, which is used for script or code generation, is derived from this name using the model naming conventions.
5. [optional] Click the *Select Extensions* button and attach one or more extensions to your model.
6. Click *OK* to create and open the enterprise architecture model .

> **Note**
>
> Sample EAMs are available in the Example Directory.

1.2.1 EAM Properties

You open the model property sheet by right-clicking the model in the Browser and selecting *Properties*.

Each enterprise architecture model has the following model properties:

Property	Description
Name/Code/Comment	Identify the model. The name should clearly convey the model's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the model. By default the code is auto-generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Filename	Specifies the location of the model file. This box is empty if the model has never been saved.
Author	Specifies the author of the model. If you enter nothing, the Author field in diagram title boxes displays the user name from the model property sheet Version Info tab. If you enter a space, the Author field displays nothing.
Version / Repository	Specify a user-defined version name and the read-only repository version number of the model. You can control which of these version values is displayed in a diagram Title symbol through the display preferences for the Title symbol.
Default diagram	Specifies the diagram displayed by default when you open the model.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

1.2.2 Setting Model Options

You can set EAM model options by selecting *Tools* > *Model Options* or right-clicking the diagram background and selecting *Model Options*.

You can set the following options on the Model Settings page:

Option	Description
Name/Code case sensitive	Specifies that the names and codes for all objects are case sensitive, allowing you to have two objects with identical names or codes but different cases in the same model. If you change case sensitivity during the design process, we recommend that you check your model to verify that your model does not contain any duplicate objects.
Enable links to requirements	Displays a Requirements tab in the property sheet of every object in the model, which allows you to attach requirements to objects (see *Requirements Management*).

Option	Description
External Shortcut Properties	Specifies the properties that are stored for external shortcuts to objects in other models for display in property sheets and on symbols. By default, *All* properties appear, but you can select to display only *Name/Code* to reduce the size of your model. **Note** This option only controls properties of external shortcuts to models of the same type (PDM to PDM, EAM to EAM, etc). External shortcuts to objects in other types of model can show only the basic shortcut properties.

For information about controlling the naming conventions of your models, see *Core Features Guide* > *Modeling with PowerDesigner* > *Objects* > *Naming Conventions*.

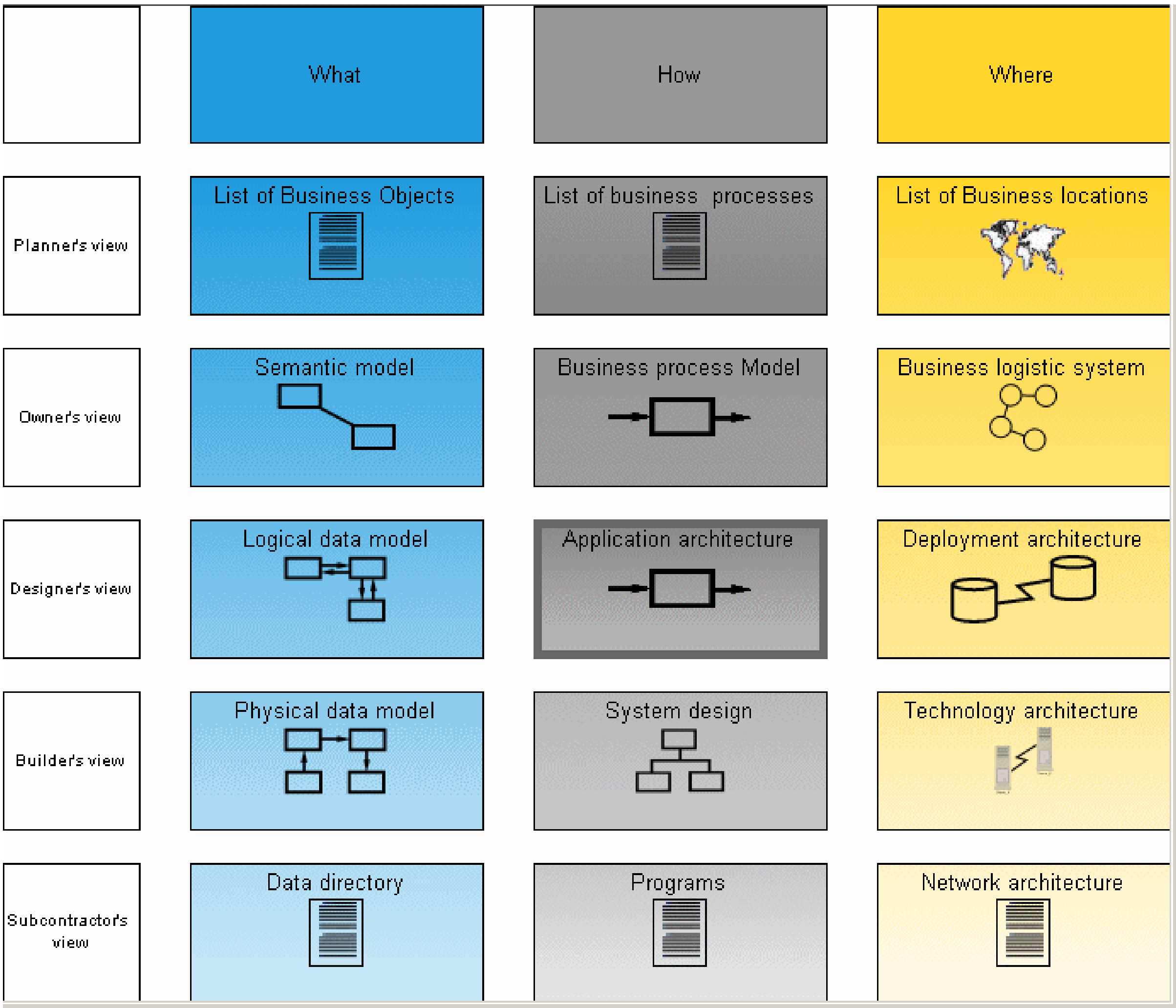
1.3 Creating an Enterprise Architecture Framework Project

PowerDesigner provides support for some of the most popular enterprise architecture frameworks in the form of project templates.

Procedure

1. Select *File* > *New Project* to open the New Project dialog.
2. In the Project type pane, expand the EA Frameworks node and select one of the available frameworks.
3. Specify a name for the project and a location to save it, and then click OK to create the project.

   The project opens to the framework matrix, from which you can create models, diagrams, lists, and other items to complete the requirements of the framework:



Results

For detailed information about working with projects, see *Core Features Guide* > *Modeling with PowerDesigner* > *Projects and Frameworks*.

1.4 Creating EAM Diagrams

When you want to visualize a part of your environment for yourself or a stakeholder, create a diagram and add the appropriate objects to it. If another stakeholder requires a different level of detail or different information, then create a separate diagram for them.

> **Note**
>
> Every PowerDesigner object can be displayed in multiple diagrams to show it in different contexts. All of the links that you create to other objects in all such diagrams are stored in the object to allow you to analyze the entirety of its dependencies (see *Core Features Guide > Linking and Synchronizing Models > Impact and Lineage Analysis*).

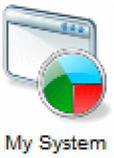
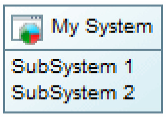
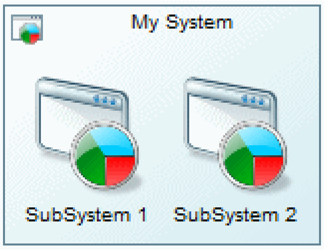
You specify a first diagram when you create an EAM. To add further diagrams in your model, right-click its Browser entry and select *New* > `<Diagram>`. The following types of diagrams are available:

- A process map provides a graphical view of your business architecture, and helps you identify your business capabilities and high-level processes, independent of the people and business units who fulfill them. See Process Maps [page 52].
- A city planning diagram provides a graphical view of the big picture of your enterprise architecture, using the metaphor of planning the infrastructure of a city to represent the organization of capabilities, systems, applications, etc into areas. See City Planning Diagrams [page 54].
- An organization chart provides a graphical view of your organization as a tree structure, and helps you analyze and display the relationships between organization units (divisions, groups, teams, etc), individuals, and roles. See Organization Charts [page 36].
- A business communication diagram provides a graphical view of your organization, and helps you analyze, the relationships, flows, and other connections between business capabilities, organization units, roles, and sites. See Business Communications Diagrams [page 37].
- An application architecture diagram provides a high-level graphical view of the application architecture, and helps you identify applications, sub-applications, components, databases, services, etc, and their interactions. See Application Architecture Diagrams [page 66].
- A service-oriented diagram provides a graphical view of your business and application services and the relationships between them, and helps you associate applications and other application layer objects with business services and processes to assist with SOA design. See Service-Oriented Diagrams [page 68].
- A technology infrastructure diagram provides a high-level graphical view of the physical architecture required to support the application architecture. See Technology Infrastructure Diagrams [page 87].
- A timeline diagram provides a calendar-style overview of your enterprise architecture programs and projects in relation to your goals, or to the assets that they impact or are in other ways related to. See Timeline Diagrams [page 99].

1.4.1 Displaying EAM Objects in Diagrams

You can create EAM objects directly in a diagram using the Toolbox or *Model* menu. You can display existing objects in a diagram by dragging them from the Browser or selecting *Symbol* > *Show Symbol* and choosing them from the various sub-tabs of the *Show Symbols* dialog.

The following symbol modes are available:

Mode	Details
Icon: 	Most EAM objects support icon or picture symbols that display a name or other limited information beneath them. To switch between icon and detail mode, right-click the symbol and toggle the *Show Detail* command or press *CTRL+Q*. To change any object symbol to an icon, select a *Custom Shape* of type Bitmap in the Symbol Format or Display Preferences dialog.
Detail: 	All objects support detail symbols (rectangles or other shapes) that can contain extensive information about the object inside the symbol, including lists of sub-objects. To control the information displayed, use the *Content* tab in the Symbol Format or Display Preferences dialog.
Editable Composite: 	Some EAM objects (see below) support editable composite symbols that allow you to create or drag and drop sub-objects on the symbol. To switch between composite and detail mode, right-click the symbol and use the *Composite View* submenu.

- To change the format or content of one or more individual symbols, select them, then right-click in the selection, and select *Format*. Use the commands *Get Format* and *Apply Format* to copy the format of one symbol to another.
- To change the format or content of all the symbols in a diagram, select *Tools* > *Display Preferences* .
- To arrange and manipulate groups of symbols, use the *Symbol* menu.

1.4.2 Setting EAM Display Preferences

PowerDesigner display preferences allow you to customize the format of object symbols, and the information that is displayed on them. To set enterprise architecture model display preferences, select *Tools* > *Display Preferences* or right-click the diagram background and select *Display Preferences*.

In the *Display Preferences* dialog, select the type of object in the list in the left pane, and modify its appearance in the right pane.

You can control what properties it will display on the *Content* tab, and how it will look on the *Format* tab. If the properties that you want to display are not available for selection on the *Content* tab, click the *Advanced* button and add them using the *Customize Content* dialog.

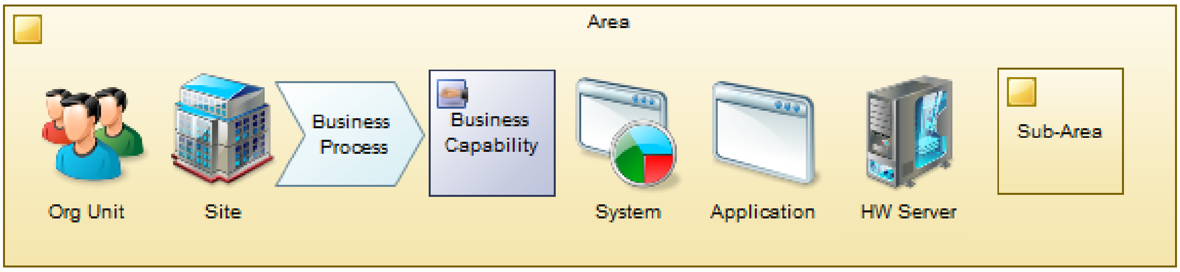
For detailed information about controlling the appearance and content of object symbols, see *Core Features Guide > Modeling with PowerDesigner > Diagrams, Matrices, and Symbols > Display Preferences*.

1.5 Structuring EAM Objects

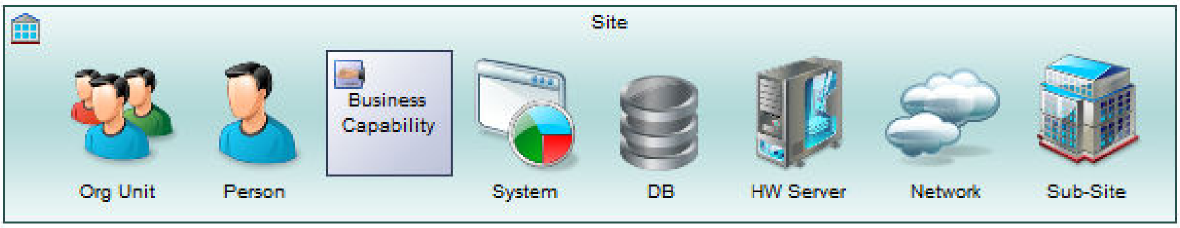
You can structure and organize your objects in various ways. Many enterprise architecture objects can be decomposed into sub-objects, and some can contain other objects and/or can be used to group other objects.

The following objects provide the principal ways for grouping other objects:

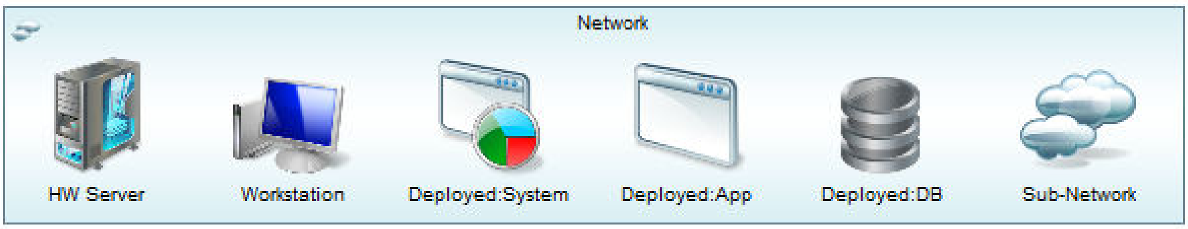
- Areas - Provide logical groupings of objects (see Areas (EAM) [page 55]). Any object can be attached to an area and can appear inside the area symbol. Objects can be attached to multiple areas:



- Sites - Provide geographical groupings of objects (see Sites (EAM) [page 39]). Many types of objects can be attached to a site and can appear inside the site symbol. Organization units, roles, business capabilities, systems, and databases can be attached to multiple sites, while people and machines can only be attached to one:

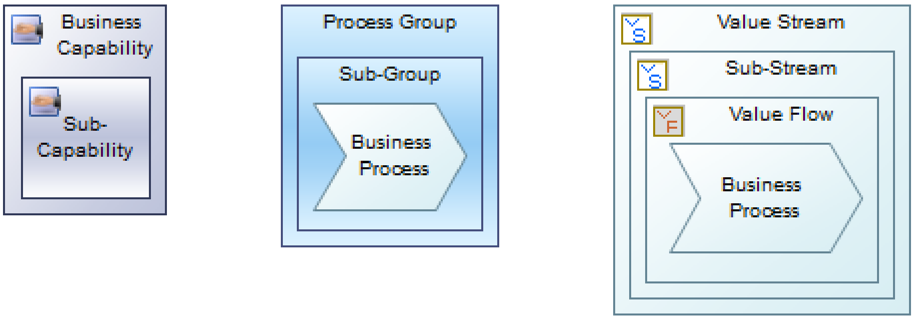


- Networks - Provide virtual groupings of objects (see Networks (EAM) [page 91]). Hardware objects and deployment instances can be attached to a network and can appear inside the network symbol:

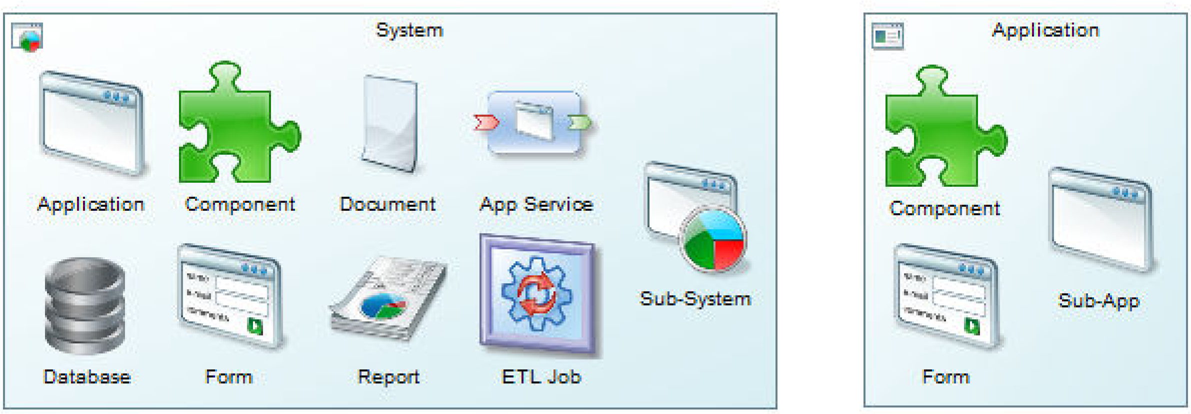


> **Note**
>
> Objects associated with an area, site, or network do not belong to these objects, and will not be deleted if you delete the area, site or network. However, areas can contain sub-areas, sites can contain sub-sites, and networks can contain sub-networks, and these sub-objects do belong to their parents and will be deleted if their parent is deleted.

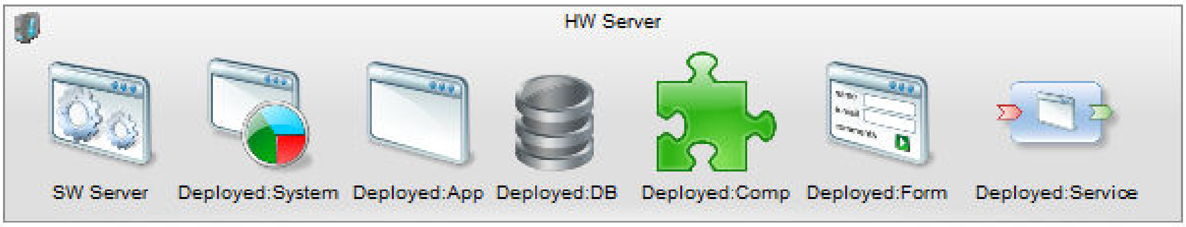
- Organization units - Can contain sub-units and can group people (see Organization Units (EAM) [page 41]). A person can belong to multiple organization units.
- Business capabilities can contain sub-capabilities (see Business Capabilities (EAM) [page 56]), process groups can contain sub-groups and processes (see Process Groups and Business Processes (EAM) [page 58]), and value streams can contain sub-streams and value flows (see Value Streams and Value Flows (EAM) [page 62]). In each case, sub-objects will be deleted if their parent is deleted. Processes can be displayed in the value flows to which they contribute value, but they do not belong to these flows and will not be deleted if you delete the flow:



- Systems and applications can contain various other application architecture objects (see Systems and Applications (EAM) [page 71]), which will be deleted if you delete the parent object:

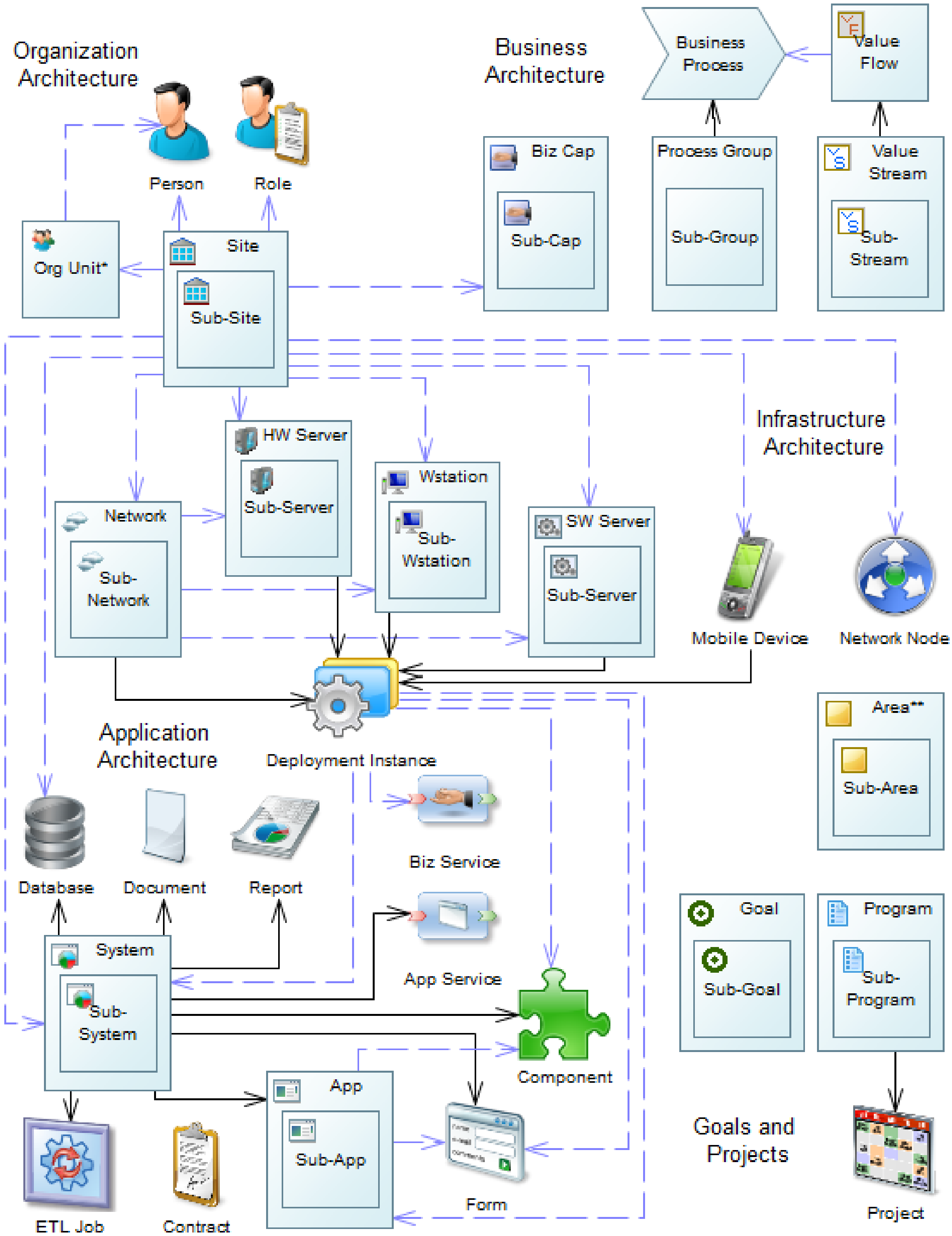


- Hardware and software servers, workstations, and mobile devices can have application architecture objects deployed to them (see Servers, Workstations, Mobile Devices, and Network Nodes (EAM) [page 89]). The deployment is modeled with deployment instances (see Deployment Instances (EAM) [page 93]). Each deployment instance is a sub-object of its parent hardware object, and will be deleted if you delete the parent, leaving the software asset itself unaffected.



In this diagram, all the main ways of decomposition, containing, and grouping objects are represented as follows:

Symbol	Meaning
Box-in-box	Decomposition - The object can be decomposed into sub-objects, which belong to the parent object and will be deleted if the parent is deleted. For example, an area can contain sub-areas and a site can contain sub-sites. Sub-objects are displayed under the parent object in the Browser and on a dedicated tab in its property sheet. **Note** *Organization units can contain sub-units but this relationship can only be shown in an organization chart.
Solid arrow	Containing sub-objects - The object can contain other objects, which belong to the parent object and will be deleted if the parent is deleted. For example, a system can contain sub-systems, applications, data-bases, and other software artifacts. Child objects are displayed under the parent object in the Browser and on dedicated tabs in its property sheet.
Dashed arrow	Grouping objects - The object can group other objects, but these grouped objects do not belong to the grouping object and will not be deleted if the grouping object is deleted. For example, a site can group peo-ple, capabilities, systems, servers, and various other objects. Grouped objects are not displayed under the grouping object in the Browser but are listed in various ways in its property sheet. **Note** **Areas can group any artifact.



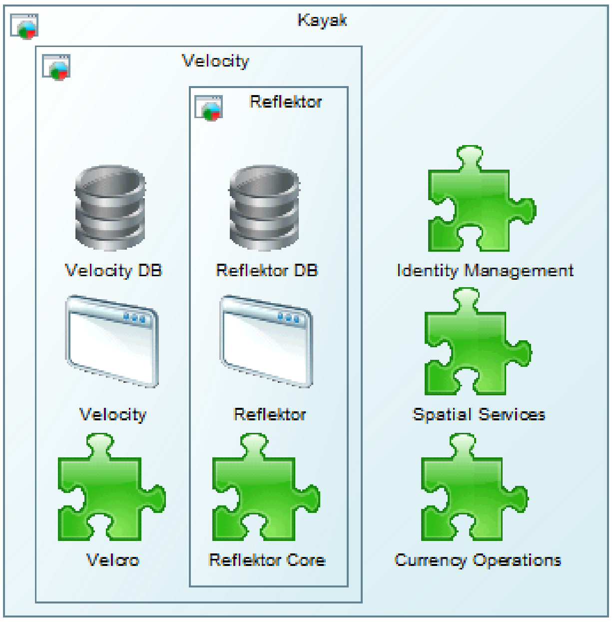
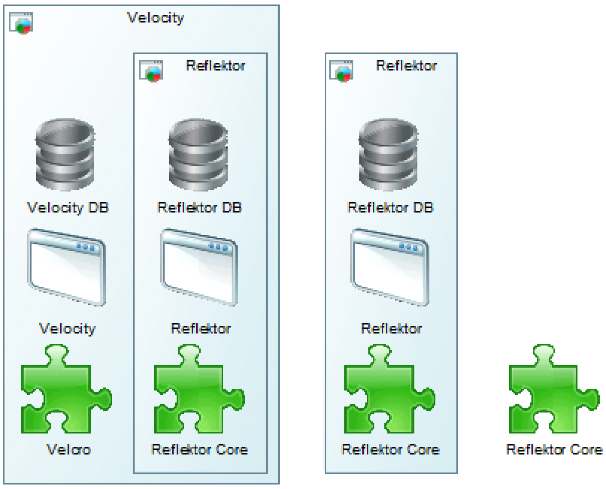
Organization Architecture
Person
Role
Org Unit*
Site
Sub-Site
Business Architecture
Business Process
Value Flow
Biz Cap
Sub-Cap
Process Group
Sub-Group
Value Stream
Sub-Stream
Infrastructure Architecture
HW Server
Sub-Server
Wstation
Sub-Wstation
SW Server
Sub-Server
Network
Sub-Network
Mobile Device
Network Node
Area**
Sub-Area
Application Architecture
Deployment Instance
Database
Document
Report
Biz Service
App Service
System
Sub-System
Component
App
Sub-App
Form
ETL Job
Contract
Goal
Sub-Goal
Program
Sub-Program
Goals and Projects
Project

1.6 Replicating EAM Objects

You can create a replica of an object from one model to another (or from one package to another) through drag and drop, or from the Replicate Objects dialog. The replica is a local copy of the object, all its properties (including replicas of all its sub-objects) are by default synchronized with its source object, and you can select to edit any or all of the properties and sub-objects locally, while keeping the others synchronized. Reusing objects in this way prevents unneccessary duplication of information and improves the quality and scope of impact analysis.

To replicate an object, right-click it in the Browser or a diagram, drag and drop it to where you want to create the replica, release the right mouse button, and select *Replicate Here*. Alternately, select *Edit* > *Replicate Objects* in the model in which you want to create the replica to open the *Replicate Objects* dialog, and select the objects to replicate. For detailed information about working with replicas, see *Core Features Guide > Linking and Synchronizing Models > Shortcuts and Replicas*.

When working with composite objects (such as hierarchies of sub-capabilities inside an EAM capability), you create a replica of any object at any level in the hierarchy, and the replica symbol will display all levels of sub-objects by default. In the following example, objects at each level of the **`Kayak > Velocity > Reflektor > Reflektor Core`** hierarchy of objects are replicated and display their sub-objects in the resulting replica symbols:

Source Object	Replicas
	

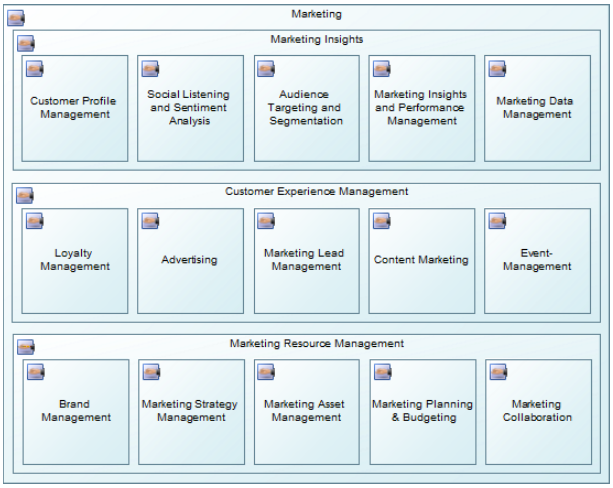
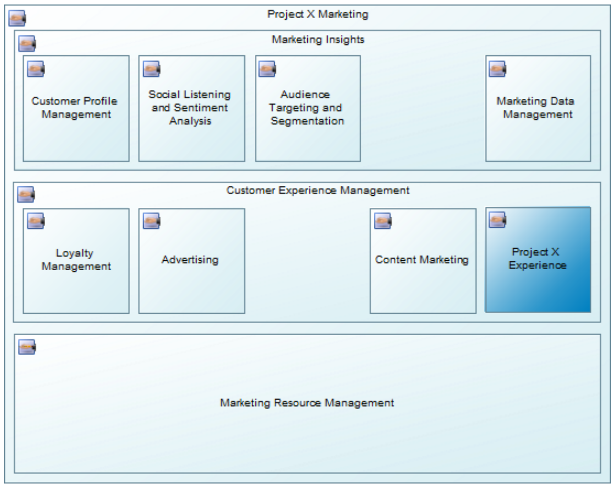
If you replicate an object by dragging and dropping a symbol from one diagram to another as a replica, the replicated object symbol will respect the formatting and content of the source symbol, including displaying only those sub-objects that are displayed on the source-symbol, and you can:

- Change the format of the replica symbol as if it were a local object, including controlling whether the composite view is shown at each level, and hiding individual sub-objects.
- Desynchronize collections of sub-objects to make them editable, and then delete sub-objects from the replica by selecting them and clicking `Delete` and create new sub-objects using the Toolbox or property sheet.

- Review and obtain any changes to desynchronized collections on the source object in your replica by right-clicking the replica in the Browser or diagram and selecting *Update Replica from Source Object*. Changes to any synchronized (non-editable) properties and collections continue to be pushed automatically to the replica.

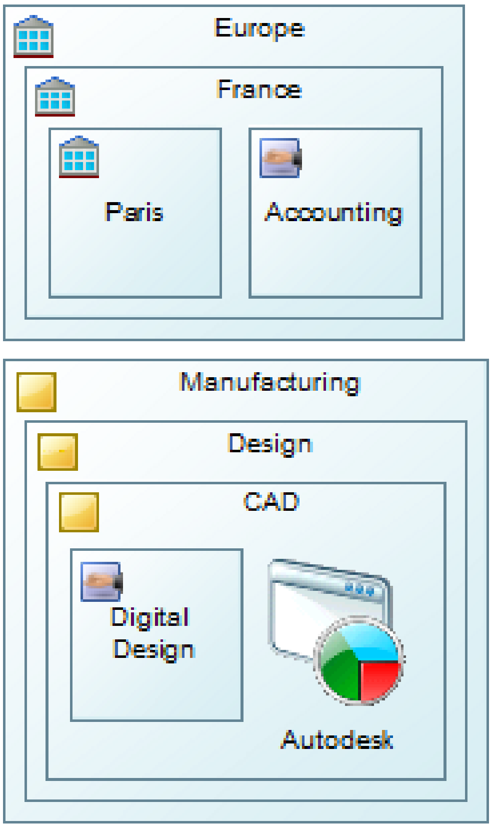
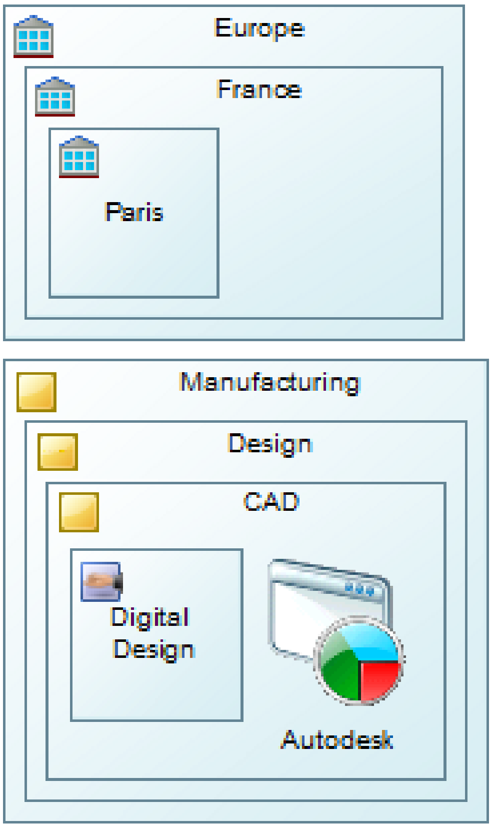
In the following example, the `Marketing` business capability is replicated and the following modifications made to the replica:

- The *Name* property has been desynchronized and changed.
- The `Marketing Insights and Performance Management` sub-sub-capability has been hidden.
- The `Customer Experience Management` *Sub-Capabilities* collection has been desynchronized, two sub-capabilities have been deleted, and `Project X Experience` has been added.
- The `Marketing Resource Management` symbol has been modified to hide its sub-capabilities by right-clicking it and selecting *Composite View* > *None*.

Source Object	Replicas
	

Only objects that are children of the composite object are replicated with it, and objects that are merely associated with it are not replicated. An exception is made for the objects associated with an area, which while not strictly sub-objects of the area, are nonetheless replicated with the area.

In the following example, the hierarchy of sites is replicated (though the business capability associated with `France` is not), while all the objects associated with the area are replicated with it:

<table>
<tr><th>Source Objects</th><th>Replicas</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

1.7 The EA Example Model, Excel Imports, and Charts

The PowerDesigner enterprise architecture metamodel is very open and can be used to model your artifacts in many different structures. To help you understand the EA metamodel and to orient you in planning your own models, we provide the EA Example model.

EA Example Files

The EA Example model and its associated files are available in the `<install_dir>/Examples/EAExample` folder. The following files are provided:

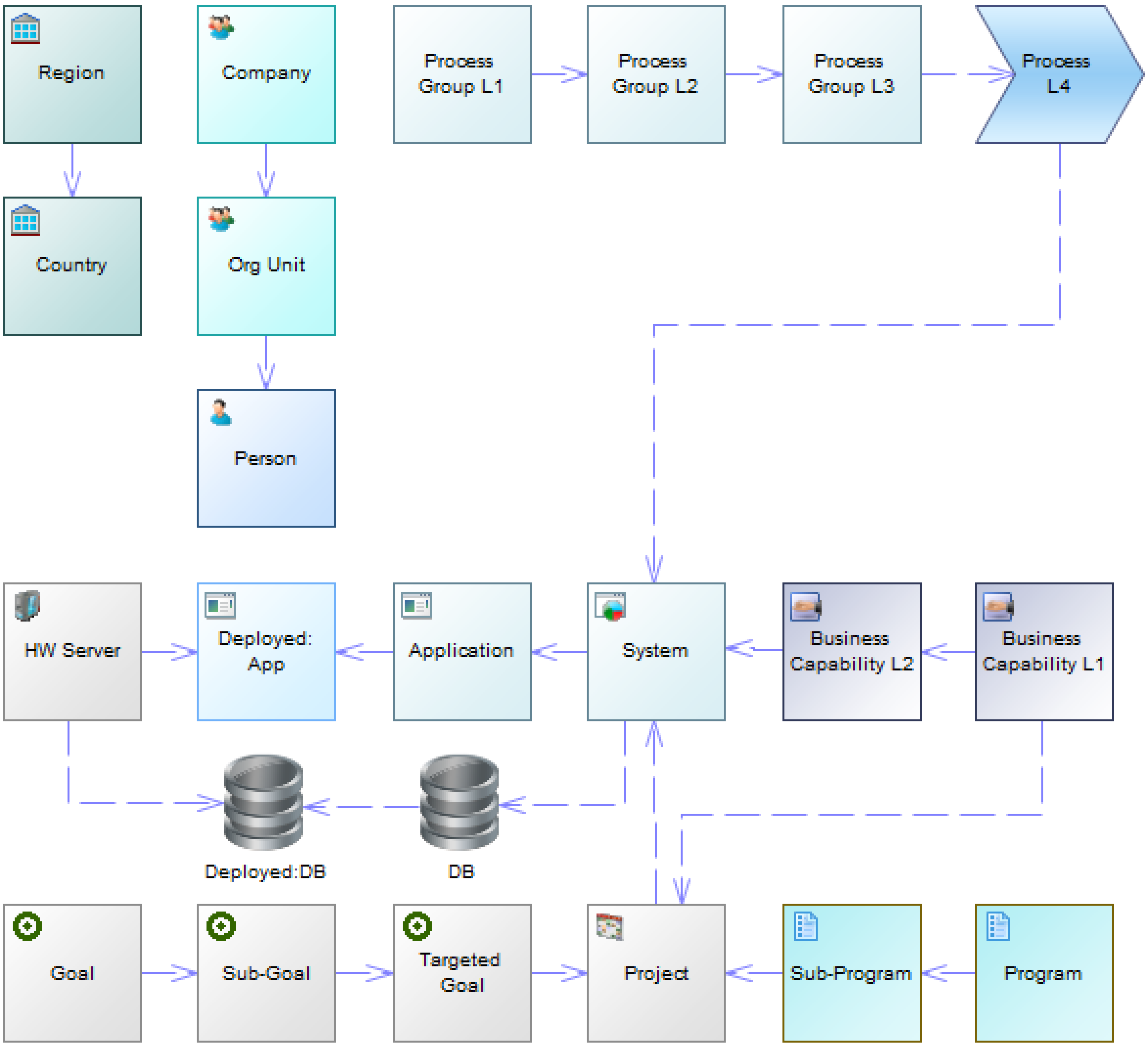
- A set of Excel files containing metadata to import (see Importing EA Metadata from Excel [page 140]) to create all the objects in EA Example, and which you can adapt to import your own artifacts.
- The `EAExample.eam` model, containing a selection of diagrams visualizing certain of these objects (see EA Example Diagrams [page 27]), which you can open in the PowerDesigner desktop client.
- The `EAExample.xem` extension file, which contains dataset definitions to drive charts to analyze the metadata contained in the model (see EA Example Charts [page 30]). This file is loaded by default in your repository

`Library` folder and can be used directly with your own models if they conform to the `EAExample.eam` structure or be adapted to your chosen structure.

- The `EAExample.db` SQL Anywhere repository database, which contains the model, extension file, and charts for viewing in a Web browser using the PowerDesigner Web client. For information about installing and accessing this repository, see *Installation Guide > Installing the Repository > Installing the EA Example Repository* .

EA Example Structure

The Acme Group created the EA Example model to understand its internal architecture and to plan changes to it by capturing metadata in PowerDesigner. After exploring the PowerDesigner metamodel, the architects decided to import a subset of the artifacts available and to structure them as follows:

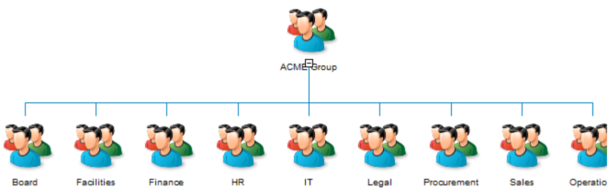
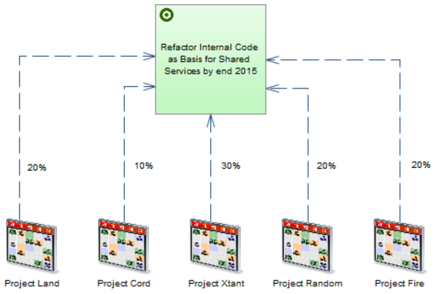


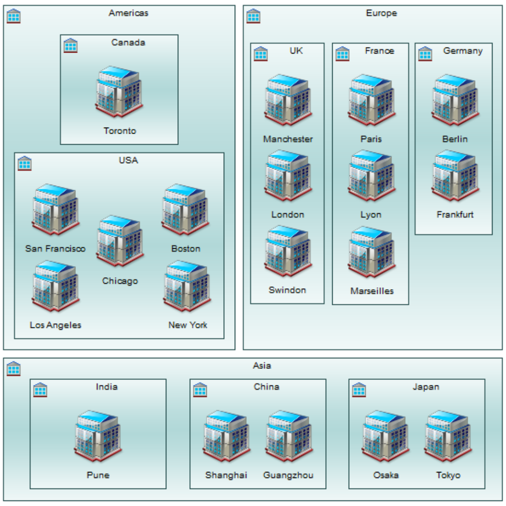
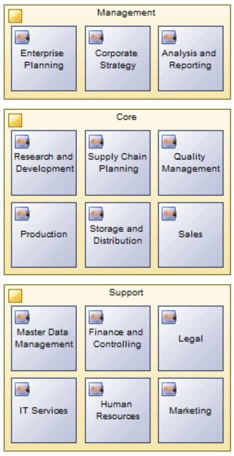
- Organization architecture - models the people, sites, and structures of the organization. EA Example contains:

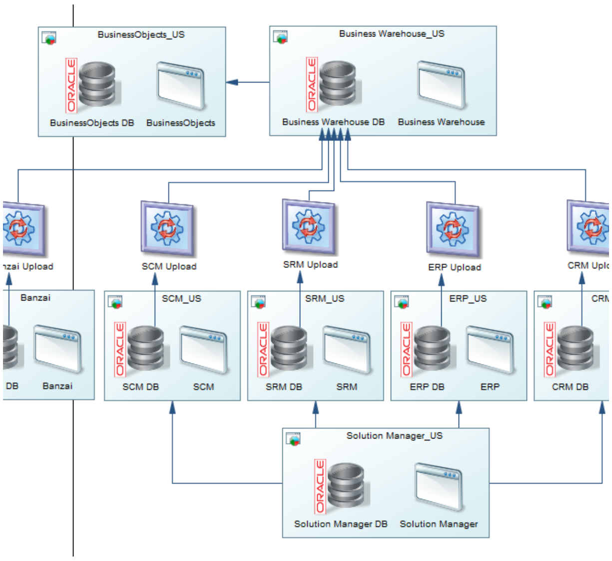
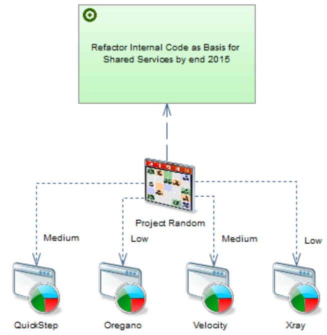
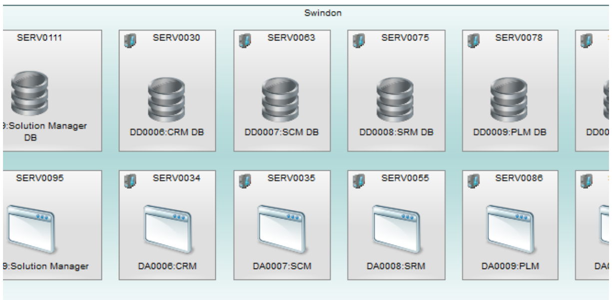
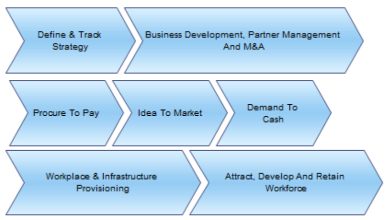
    - Sites - in three levels, from regions, through countries, to sites.
    - Organization Units - in two levels, with people belonging to the second-level units and associated with a site.
- Process and capability architecture - models what the organization does and how it does it. EA Example contains:
    - Business Capabilities - in two levels, with first-level capabilities linked to projects and second-level capabilities linked to systems via traceability links.
    - Process Groups - in three levels.
    - Business Processes - as a fourth level under the process groups, linked to systems via traceability links.
- Application architecture- models the software that the organization uses. EA Example contains:
    - Systems - which can contain applications and databases.
- Infrastructure architecture - models the hardware and physical assets that the organization uses. EA Example contains:
    - Hardware Servers - which are located in sites, and to which systems are deployed.
- You can model your corporate goals, how they will be fulfilled through changes to your architecture and what impacts these changes will have on your artifacts. EA Example contains:
    - Goals - in three levels, with third-level goals fulfilled by projects.
    - Programs - in two levels, with second-level programs containing projects.
    - Projects - which impact systems.
- In addition to the links shown in the diagram, EA Example contains role association links, which connect people to the following objects:
    - Business Capabilities (as owner)
    - Processes (as owner)
    - Hardware Servers (as owner)
    - Databases (as owner)
    - Applications (as owner)
    - Programs (as sponsor)
    - Projects (as sponsor)

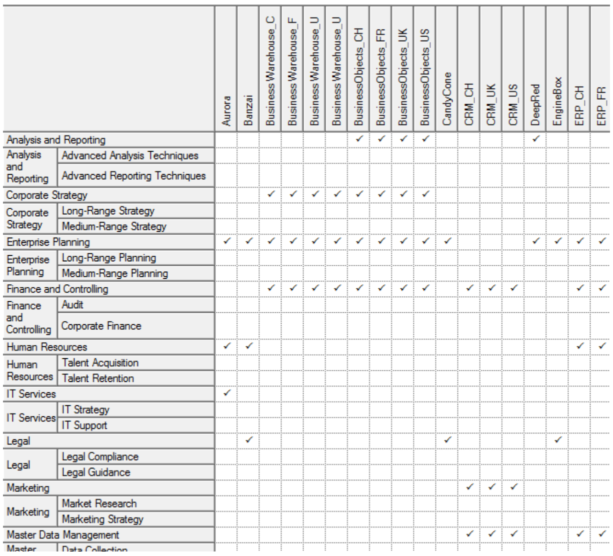
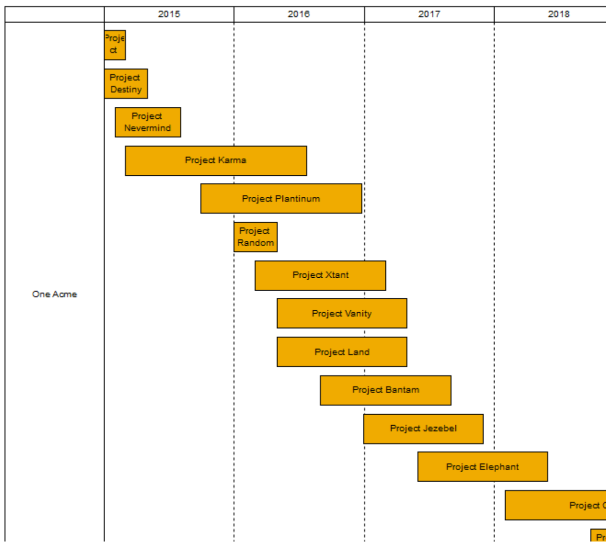
1.7.1 EA Example Diagrams

Diagrams allow you to present selected artifacts for visualization. The EA Example model contains a number of example diagrams, which can be viewed with the PowerDesigner desktop client or by installing the EA Example repository and navigating to it with PowerDesigner Web.

Description	Diagram
Organization Chart Shows the structure of organization units in Acme Group	
Corporate Goals Shows high-level corporate goals broken down to three levels	
Shared Services Projects Shows the projects that contribute to fulfilling a particular goal.	

Description	Diagram
Worldwide Sites Shows the company sites organized by region and country.	
Capabilities Shows capabilities organized by area.	

Description	Diagram
Business Warehouse US Shows the systems that upload data to this system.	
Project Random Shows the goal to which this contributes and the systems it will impact.	
Swindon Server Farm Shows the servers deployed to this site and the databases and applications deployed to them.	
Process Map Show the company's level one processes.	

Description	Diagram
Capabilities - System Dependencies Shows the systems on which level one and two business capabilities depend.	
`Core Technologies` (and other timelines) Show the timeline for scheduling the EA projects, organized by goals and other criteria.	

1.7.2 EA Example Charts

Charts allow you to visualize statistics drawn from your metadata. The EA Example model has a number of example charts defined for it, which can be viewed by installing the EA Example repository and navigating to it with PowerDesigner Web.

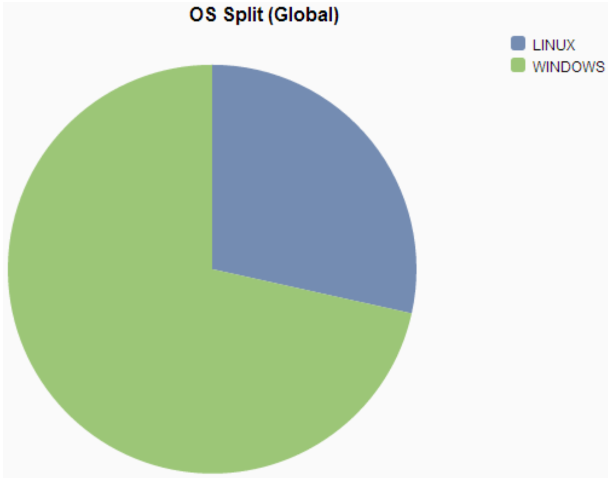
The following example charts are defined in the EA Charts extension file, which is loaded by default in your repository library (and which is delivered as part of the EA Example files available at `<install_dir>/Examples/EAExample`). For information about working with datasets, see *Customizing and Extending PowerDesigner > Extension Files > Chart Datasets (Profile)*

<table>
<tr><th>Dataset and Chart Configuration</th><th>Chart</th></tr>
<tr><td>
Question: What OS are my servers running?

Path: Model > Hardware Server (Operating System, Operating System Version)

Chart: A pie chart with each OS as a piece of the pie:

Pie Sectors: <code>Count (of hardware servers)</code>
Legend Color: <code>Operating System</code>

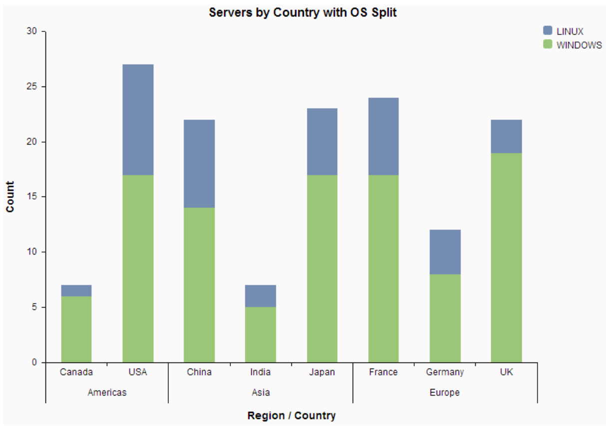
</td><td>


</td></tr>
<tr><td>
Question: What OSs are running where?

Path: Site > Site > Site > Hardware Server (Operating System, Operating System Version)

Chart: A stacked bar chart with OSs as stacked colored bars

Measures: <code>Count (of hardware servers)</code>
X Axis: <code>Region, Country</code>
Legend Color: <code>Operating System</code>

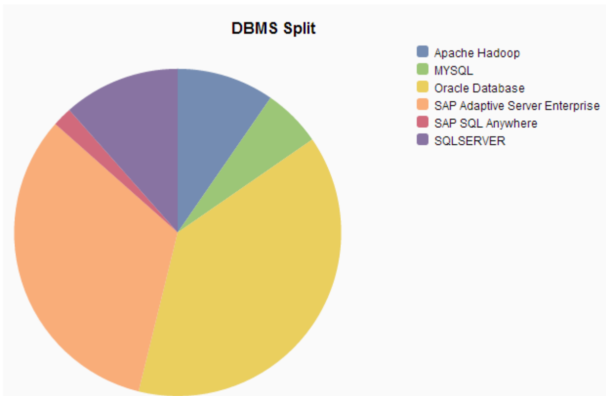
</td><td>


</td></tr>
<tr><td>
Question: What DBMSs are my servers running?

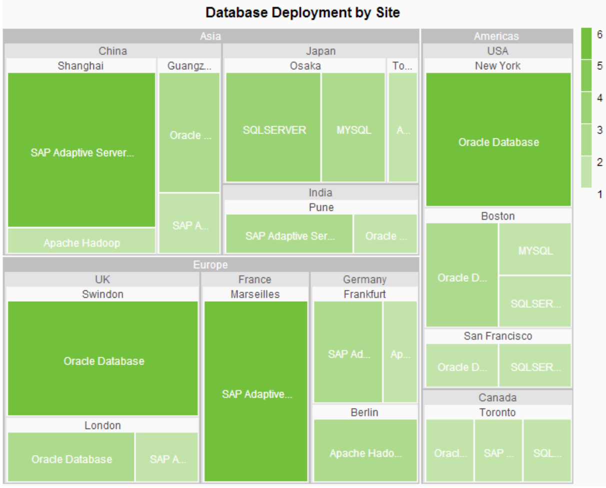
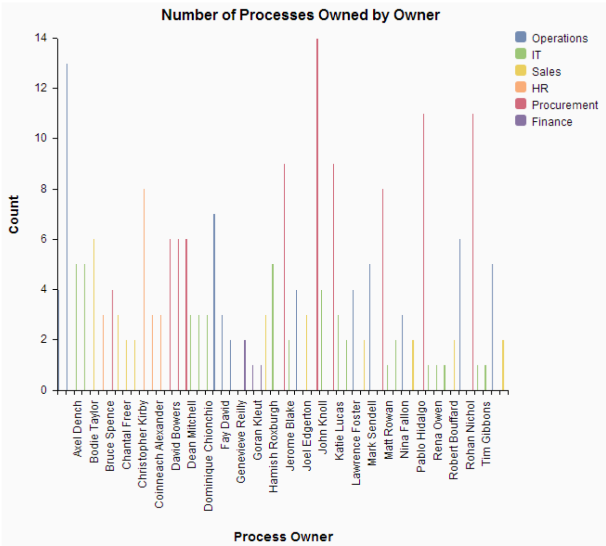
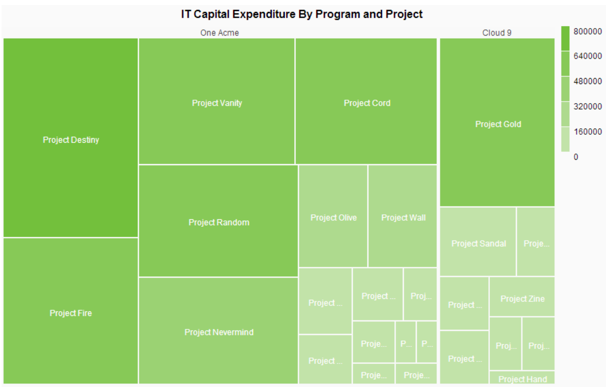
Path: Model > Database (DBMS, DBMS Version)

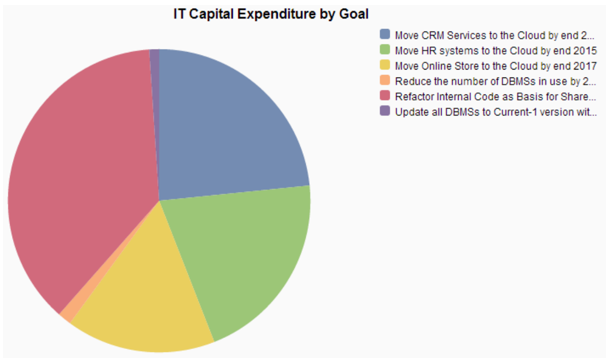
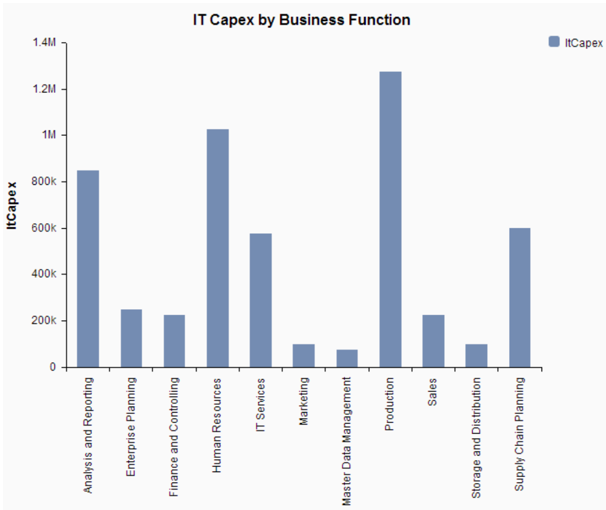
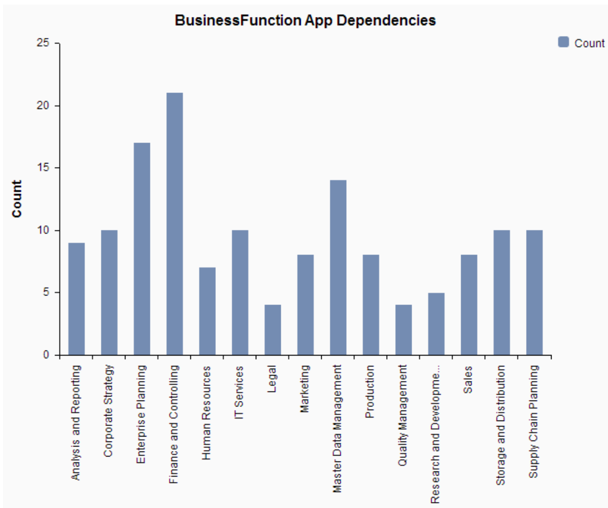
Chart: A pie chart with each DBMS as a piece of the pie:

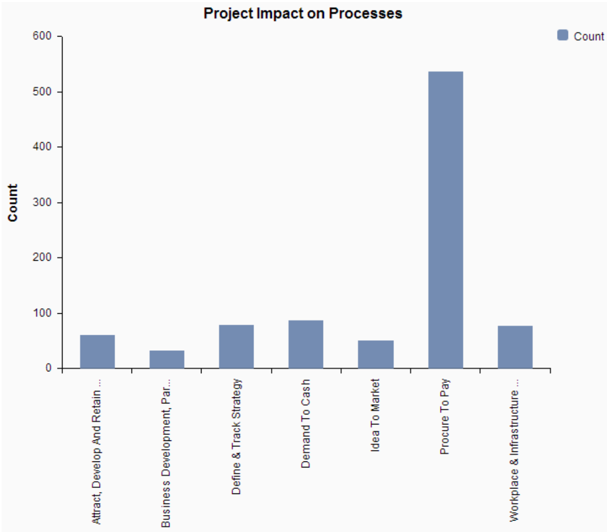
Pie Sectors: <code>Count (of databases)</code>
Legend Color: <code>DBMS</code>

</td><td>


</td></tr>
</table>

Dataset and Chart Configuration	Chart
Question: What DBMSs are deployed where? Path: *Site* › *Site* › *Site* › *Hardware Server* › *Deployment Instance* › *Database (DBMS, DBMS Version)* Chart: A tree map: • Area Weight: `Count (of databases)` • Area Color: `Count (of databases)` • Area Name: `Region, Country, Site, DBMS`	
Question: Who are my process owners and what processes do they own? Path: *Organization Unit* › *Organization Unit* › *Person* › *Role Association* › *Process* Chart: A bar chart with a bar per person: • Y Axis: `Count (of processes)` • X Axis: `Person` • Legend Color: `Organization Unit`	
Question: What is my IT Capex by Program and Project? Path: *Program* › *Program* › *Project (ItCapex)* Chart: A tree map: • Area Weight: `ItCapex` • Area Color: `ItCapex` • Area Name: `Program, SubProgram, Project`	

Dataset and Chart Configuration	Chart
Question: What is my IT Capex by Goal? Path: *Goal* > *Goal* > *Goal* > *Fulfillment* > *Project (ItCapex)* Chart: A pie chart with each goal as a piece of the pie: • Pie Sectors: `ItCapex` • Legend Color: `Goal`	
Question: What is my IT Capex by Business Capability? Path: *BusinessCapability* > *BusinessCapability* > *TraceabilityLink* > *Project (ItCapex)* Chart: A bar chart with a bar per business capability: • Y Axis: `ItCapex` • X Axis: `L1 Business Capability`	
Question: How complex are the dependencies of my business capabilities on applications? Path: *BusinessCapability* > *BusinessCapability* > *TraceabilityLink* > *System* > *EnterpriseApplication* Chart: A bar chart with a bar per business capability: • Y Axis: `Count (of applications)` • X Axis: `L1 Business Capability`	

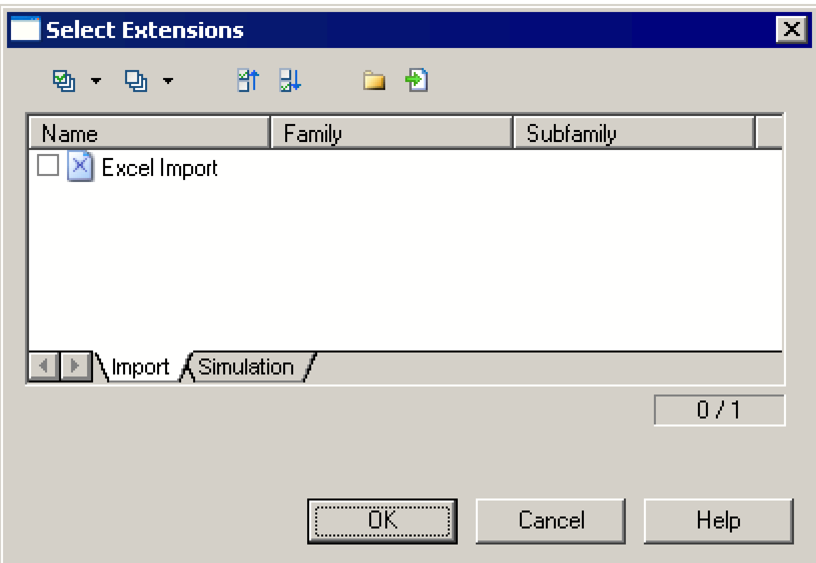
Dataset and Chart Configuration	Chart
Question: What is the impact of my projects on processes? Path: *Process* > *Process* > *Process* > *Process* > *TraceabilityLink* > *System* > *Impact* > *Project* Chart: A bar chart with a bar per business capability: • Y Axis: `Count (of projects)` • X Axis: `L1 Business Capability`	

1.8 Extending your Modeling Environment

You can customize and extend PowerDesigner metaclasses, parameters, and file generation with extensions, which can be stored as part of your model or in separate extension files (*.xem) for reuse with other models.

To access extensions defined in a `*.xem` file, simply attach the file to your model. You can do this when creating a new model by clicking the *Select Extensions* button at the bottom of the New Model dialog, or at any time by selecting *Model* > *Extensions* to open the List of Extensions and clicking the *Attach an Extension* tool.

In each case, you arrive at the Select Extensions dialog, which lists the extensions available, sorted on sub-tabs appropriate to the type of model you are working with:



To quickly add a property or collection to an object from its property sheet, click the menu button in the bottom-left corner (or press F11) and select *New Attribute* or *New List of Associated Objects*. For more information, see *Core Features Guide > Modeling with PowerDesigner > Objects > Extending Objects*.

To create a new extension file and define extensions in the Resource Editor, select *Model* > *Extensions*, click *Add a Row*, and then click *Properties*. For detailed information about working with extensions, see *Customizing and Extending PowerDesigner > Extension Files* .

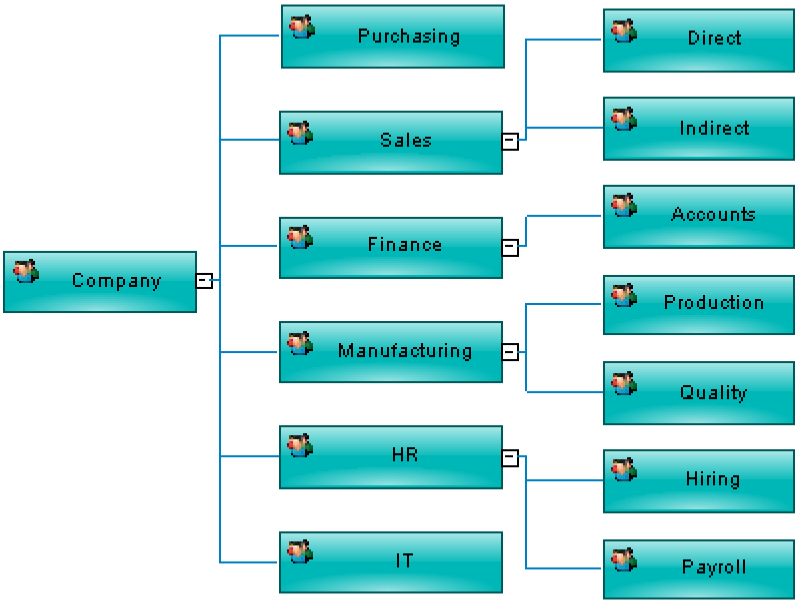
2 Organization Architecture Modeling

PowerDesigner provides you with various tools to capture, analyze, visualize, and plan changes to your organization. You can create organization charts to show the structure of your organization and business communication diagrams to show the information that transits across your sites and organization units.

2.1 Organization Charts

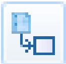
An organization chart provides a graphical view of your organization as a tree structure, and helps you analyze and display the relationships between organization units (divisions, groups, teams, etc), individuals, and roles.

In the following example, the company is analyzed into its top-level departments and the major groups within them:



To create a organization chart in an existing EAM, right-click the model in the Browser and select *New* > *Organization Chart Diagram*. To create a new model, select *File* > *New Model*, choose Enterprise Architecture Model as the model type and *Organization Chart Diagram* as the first diagram, and then click *OK*.

The following objects can be created from the organization chart toolbox:

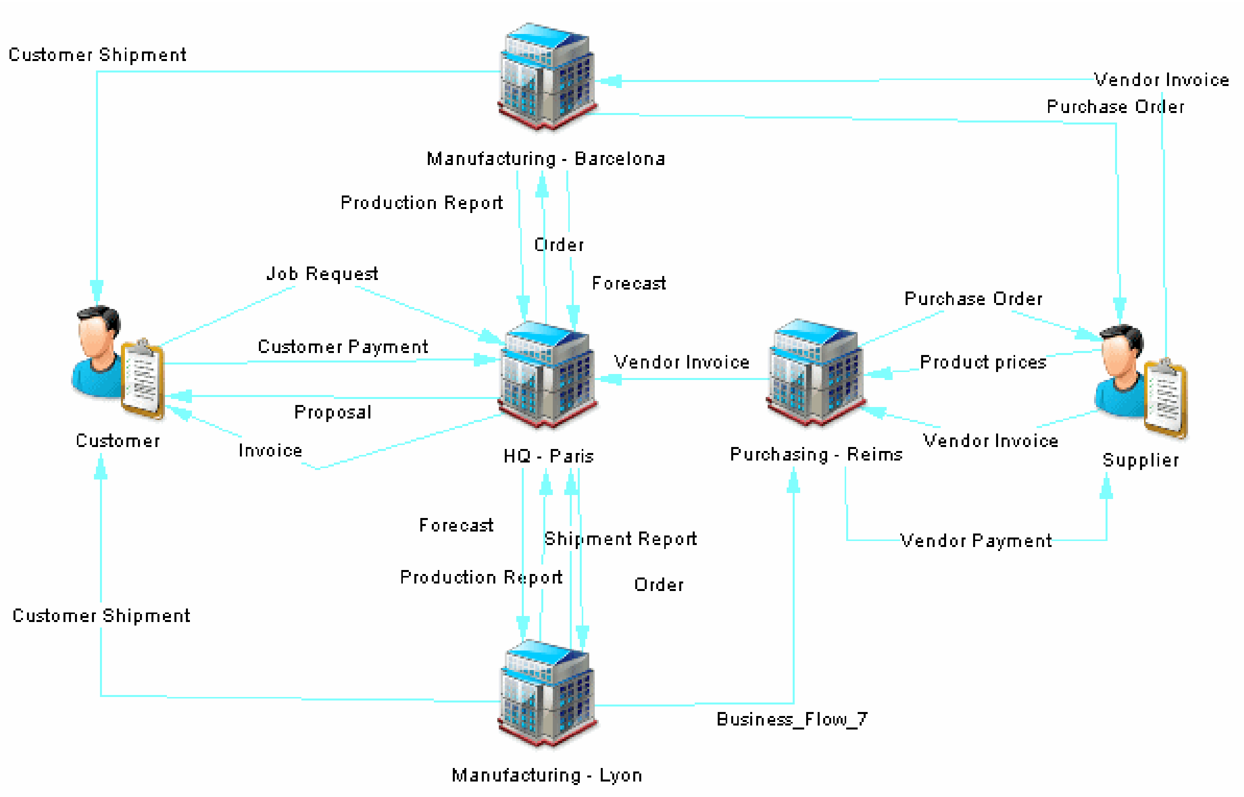
Tool	Description
	A group, department, function, or other collection of people or organization units. See Organization Units (EAM) [page 41].
	An individual. See People (EAM) [page 42].
	A link between organization units or people. See Roles (EAM) [page 47].
	A high-level EA initiative. See Programs, Projects, and Phases (EAM) [page 103]
	An EA initiative. See Programs, Projects, and Phases (EAM) [page 103]
	A mission, vision, strategy, or objective. See Goals (EAM) [page 102]
	*Impact* - A link from an EA initiative to the asset that it impacts. See Specifying the Impact of a Project on EA Assets [page 105].
	*Role Association* - A link from an EA asset to a person, group, or role that is responsible for or otherwise associated with it. See Associating a Person, Organization Unit, or Role with an Object [page 44].

For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

2.2 Business Communications Diagrams

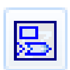
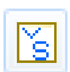
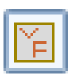
A business communication diagram provides a graphical view of your organization, and helps you analyze, the relationships, flows, and other connections between business capabilities, organization units, roles, and sites.

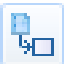
In the following example, communications between the company, its customers and suppliers are analyzed, as well as those between the different company sites:



To create a business communication diagram in an existing EAM, right-click the model in the Browser and select *New* > *Business Communication Diagram*. To create a new model, select *File* > *New Model*, choose Enterprise Architecture Model as the model type and *Business Communication Diagram* as the first diagram, and then click *OK*.

All objects can be displayed in a business communication diagram but the primary focus is on the following business objects, which can be created from the toolbox:

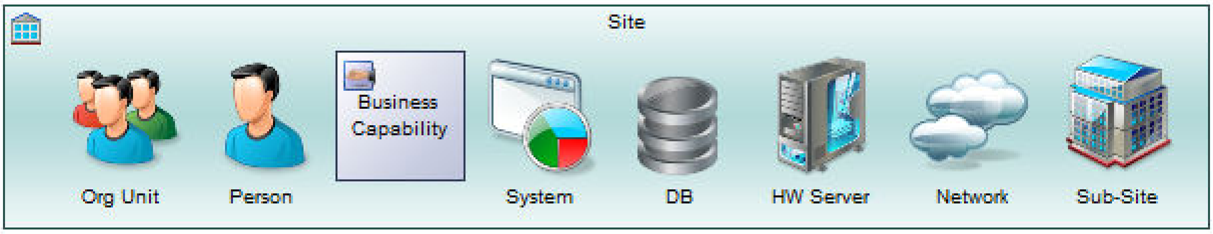
Tool	Description
	*Area* - An abstract object for grouping other objects. See Areas (EAM) [page 55].
	*Business Capability* - An aggregation of processes and/or sub-capabilities. See Business Capabilities (EAM) [page 56].
	
	*Business Process* - An activity performed in the enterprise. See Process Groups and Business Processes (EAM) [page 58].
	
	

Tool	Description
	*Person* - An individual. See People (EAM) [page 42].
	*Organization Unit* - A group, department, or other collection of people or organization units. See Organization Units (EAM) [page 41].
	*Role* - A set of responsibilities. See Roles (EAM) [page 47].
	*Site* - A physical location. See Sites (EAM) [page 39].
	*Program* - A high-level EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	*Project* - An EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	*Goal* - A mission, vision, strategy, or objective. See Goals (EAM) [page 102].
	*Business Flow* - A link between elements in a business diagram. See Business Flows (EAM) [page 48].
	*Role Association* - A link from an EA asset to a person, group, or role that is responsible for or otherwise associated with it. See Associating a Person, Organization Unit, or Role with an Object [page 44].
	*Impact* - A link from an EA initiative to the asset that it impacts. See Specifying the Impact of a Project on EA Assets [page 105].
	*Fulfillment* - A link from an EA initiative to a goal that it contributes to fulfilling. See Specifying the Fulfillment of Goals by a Project [page 104].

For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

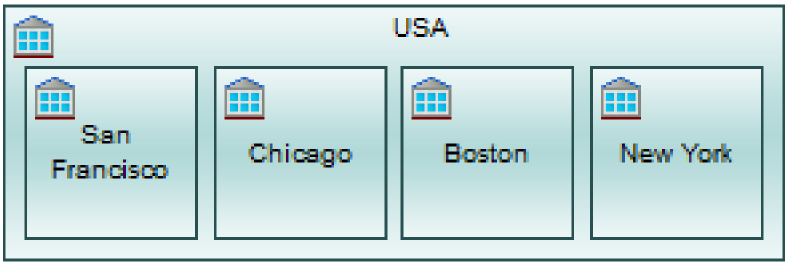
2.3 Sites (EAM)

A site is a physical location that can contain sub-sites and can geographically group people, organizations and various other objects.



For more information about grouping and organizing EAM objects, see Structuring EAM Objects [page 18].

In this example, the USA site contains the San Francisco, Chicago, Boston, and New York sub-sites:



Creating a Site

You can create a site from the Toolbox, Browser, or *Model* menu, or on the *Sub-Sites* tab of another site.

Site Properties

To view or edit a site's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Address / Zip code / City / Country	Specify the address of the site.
Phone	Specifies the telephone number of the site.
Population	Specifies the number of people at the site.

Property	Description
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

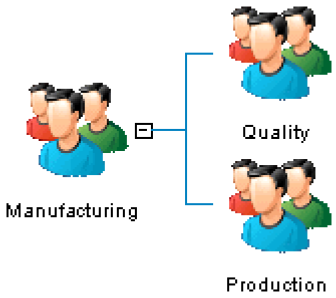
The following tabs are also available:

- Role Associations – lists the organization units, people, and roles associated with the site, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Sub-Sites – lists the sites contained within the site.

2.4 Organization Units (EAM)

An organization unit is a department, group, or team that can contain sub-units and can group people.

In this example, the Manufacturing organization unit contains the Quality and Production sub-units:



Creating an Organization Unit

You can create an organization unit from the Toolbox, Browser, or *Model* menu, or on the *Sub-Organizations* tab of another organization unit.

Organization Unit Properties

To view or edit an organization unit's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Manager	Specifies the manager of the organization unit.
Telephone	Specifies the telephone number of the organization unit.
Email	Specifies the email address of the organization unit.
Web site	Specifies the url for the web site of the organization unit.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

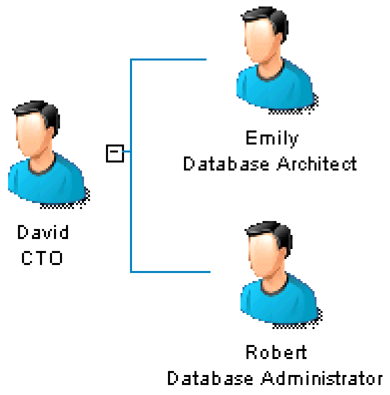
The following tabs are also available:

- Sub-Organizations – lists the organization units that belong to this organization unit.
- People – lists the people (see People (EAM) [page 42]) that are associated with this organization unit.
- Roles – lists the roles (see Roles (EAM) [page 47]) that are associated with this organization unit.
- Sites – lists the sites (see Sites (EAM) [page 39]) where this organization unit is located.

2.5 People (EAM)

A person is an individual who holds a position within an organization and plays roles in relation to artifacts.

In this example, David is the manager of Emily and Robert:



Creating a Person

You can create a person from the Toolbox, Browser, or *Model* menu, or on the *People* tab of an organization unit.

Person Properties

To view or edit a person's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Manager	Specifies the manager of the person. Select a person from the list or click the New tool to the right of the field to create a new one.
Site	Specifies the site where this person is located.
Job title	Specifies the job title of the person.

Property	Description
Telephone	Specifies the telephone number of the person.
Email	Specifies the email address of the person.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Roles – lists the roles (see Roles (EAM) [page 47]) that this person fulfills.

2.5.1 Associating a Person, Organization Unit, or Role with an Object

You can link a person, organization unit, or role to another object using the Toolbox *Role Association* tool or on the *Role Associations* tab of the property sheet of the object. The role association link is displayed in the diagram, and the list of role associations for a person, organization unit, or role is available from the *Dependencies* tab of its property sheet.

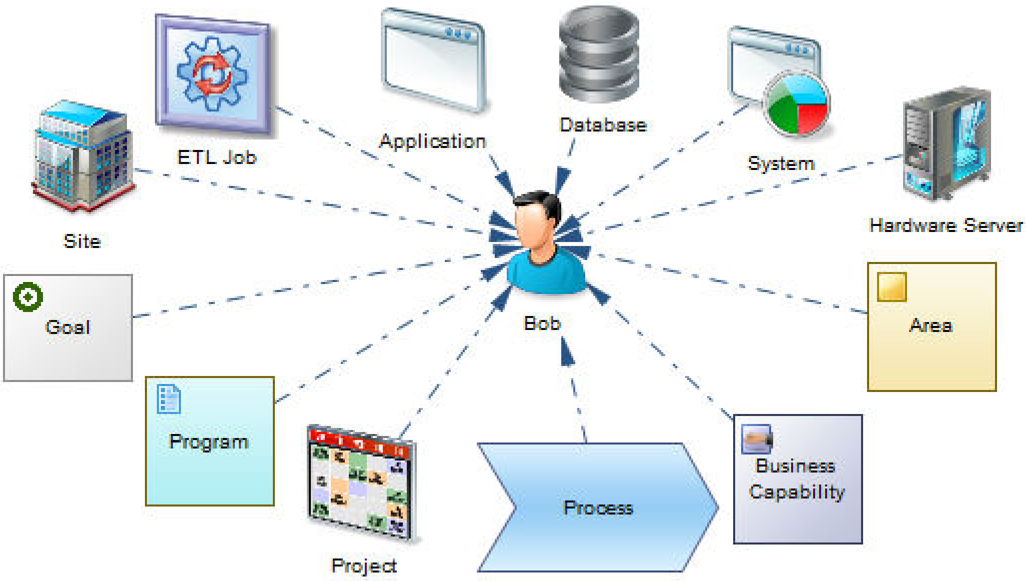
Context

For example:

- A person may be defined as the administrator of a hardware server and the manager of a site.
- An organization unit may be identified as being responsible for the Sales business capability.
- A role may be designated as the contact for the Madrid site.

To create the role association directly in an organization chart, business communication diagram, or service-oriented diagram, click the *Role Association* tool in the Toolbox, and draw a link from the object to the person, organization unit, or role. Double-click the object to open its property sheet, click the *Role Associations* tab, and specify the role type.

In this example, Bob is responsible for a wide range of objects:



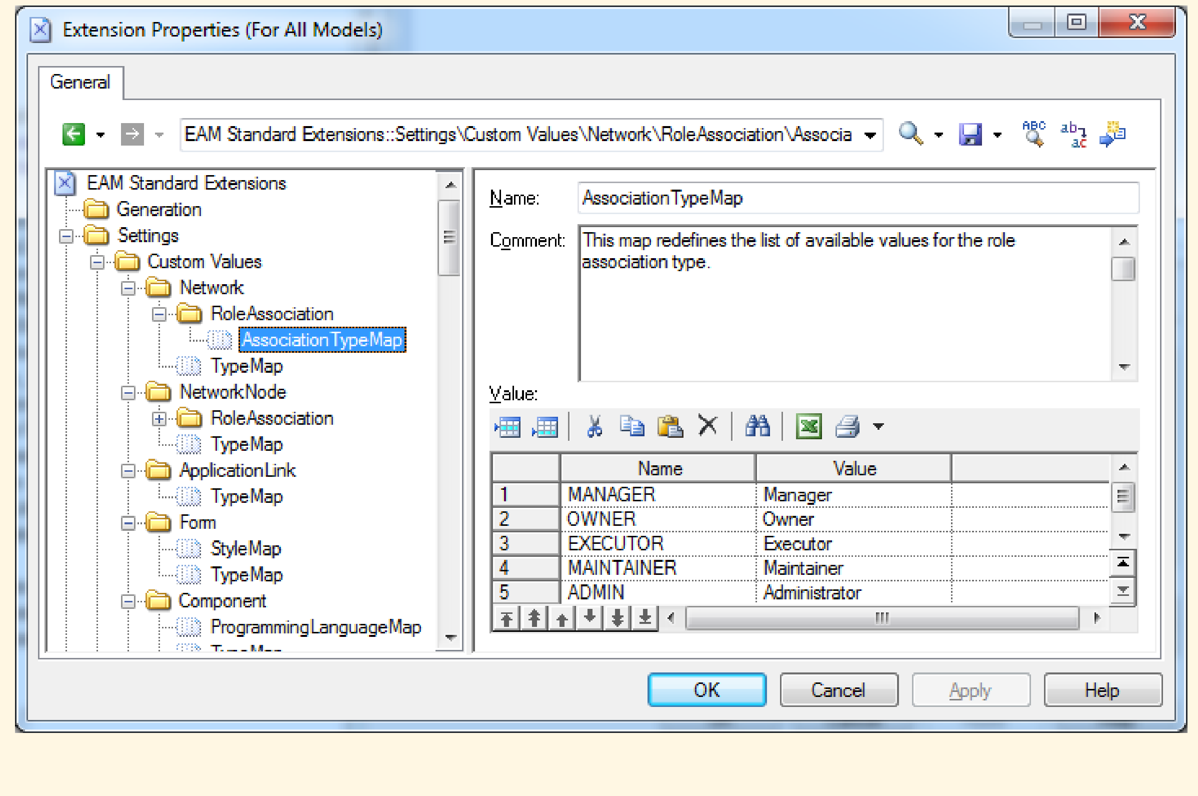
To specify role associations in any diagram, including those where the *Role Association* tool is not available:

Procedure

1. Open the property sheet of the object to which you want to connect a person, organization unit, or role, and click the *Role Associations* tab.
2. Click the *Add Objects* tool to open a selection box listing all the roles, people, and organization units available in the model, select the roles, people, and organization units that you want to associate with the object and click *OK* to return to the *Role Associations* tab.
3. [optional] Select or enter a type for each role association and then click *OK* to return to the diagram.
   If the role player and the object are in the current diagram then the link will be drawn between them.

> **Note**
>
> To display objects in a diagram, select *Symbol* > *Show Symbols* and select the relevant objects from the dialog. To display role associations, and other links in the diagram, select *Tools* > *Complete Links*.

> **Note**
>
> The default list of types for each type of object can be modified in an extension file. Navigate to `Settings/Custom Values/<object>/RoleAssociation/AssociationTypeMap` and modify the list as appropriate:



2.5.2 Importing Role Associations from Excel

You can import role associations, which are used to link people to other assets in your models, from lists in Excel files. An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File*. Select *Enterprise Architecture Model* from the model type list, enter a **`Model name`**, and then click *OK* to open the Excel Import Wizard.

> **Note**
>
> You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To

attach it, select **Model > Extensions** to open the List of Extensions, click the *Attach an Extension* tool, select **Excel Import** on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `05 Role Associations Import.xlsx` contains five sheets, each of which designates people as owners of various objects imported from precedent imports.

   > **Note**
   >
   > Certain metadata in the file cannot be automatically evaluated by the wizard. This file cannot be imported using the *Auto-map columns to properties* option and you must specify the mappings in the wizard.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:
   - All options - deselected.
   - *Reference associated object by*: **Code**
   - *Qualified name separator*: **.** (dot)

   For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.

4. Click *Next* to begin the import.

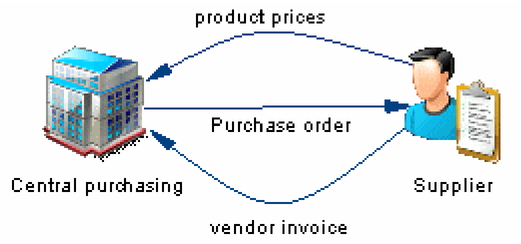
   Specify how your sheets and columns will be imported in the wizard as follows:
   - **Process Owners**:
     1. Import the table as **Business Process.Role Association**.
     2. Import the column **Parent** as **Parent**.
     3. Import the column **Role Player** as **Role Player** and set the *Object* type to **Person**.
     4. Import the column **Type** as **Type**.
   - **Capability Owners** - Import the table as **Business Capability.Role Association** and its columns in the same way as for **Process Owners**.
   - **Server Owners** - Import the table as **Hardware Server.Role Association** and its columns in the same way as for **Process Owners**.
   - **Application Owners** - Import the table as **Application.Role Association** and its columns in the same way as for **Process Owners**.
   - **Database Owners** - Import the table as **Database.Role Association** and its columns in the same way as for **Process Owners**.

   Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

2.6 Roles (EAM)

A role is a set of responsibilities. Roles can be assigned to people or organization units or can be used in place of them.

In this example, the Supplier role interacts with the Central Purchasing site:



Creating a Role

You can create a role from the Toolbox, Browser, or *Model* menu.

Role Properties

To view or edit a role's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

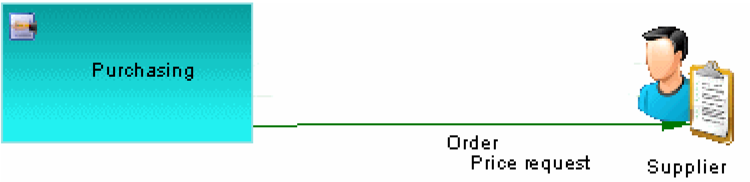
The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

2.7 Business Flows (EAM)

A business flow is an oriented link that is used to connect objects, and which can contain documents.

In this example the Price request flow goes from the Purchasing capability to the Supplier role:



Creating a Business Flow

You can create a business flow from the Toolbox, Browser, or *Model* menu.

Business Flow Properties

To view or edit a business flow's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
First/ Second object	Specify the objects at the origin and endpoint of the link.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Documents – lists the documents associated with the flow (see Documents, and Reports (EAM) [page 76]).
- Condition – lets you specify a condition to be evaluated to determine if the flow will be triggered, as well as a short alias for the condition, which can be displayed on the flow symbol.

2.8 Importing Organization Architecture Metadata from Excel

You can import your site and organization structure and the people necessary for your models from lists in Excel files. An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File*. Select *Enterprise Architecture Model* from the model type list, enter a **`Model name`**, and then click *OK* to open the Excel Import Wizard.

   > **Note**
   >
   > You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To attach it, select *Model* > *Extensions* to open the List of Extensions, click the *Attach an Extension* tool, select **`Excel Import`** on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `01 Organization Architecture Import.xlsx` contains three sheets:
   - **`Site`** - Imports a three-level structure of sites (regions, countries, and city sites). The **`Parent`** column contains the code of the site to which this site belongs so that, for example, the site France (FR) is a child of the site Europe (EU), and the site Paris (PA) is a child of the site France (EU.FR). The full path to the site is stored in the **`Comment`**
   - **`OrganizationUnit`** - Imports a two-level structure of organization units, where the various divisions all belong to a single **`Parent`**.
   - **`Person`** - Imports a number of people, each of whom is assigned to an organization unit, a manager, and a site.

   > **Note**
   >
   > As all the necessary metadata is contained within the file, selecting the *Auto-map columns to properties* allows the wizard to import all the objects without further intervention.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:

   - *Auto-map columns to properties* (selected)
   - All other options - deselected.

- *Reference associated object by*: **`Code`**
- *Qualified name separator*: **`.`** (dot)

For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.

4. Click *Next* to begin the import.

   Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

3 Business Architecture Modeling

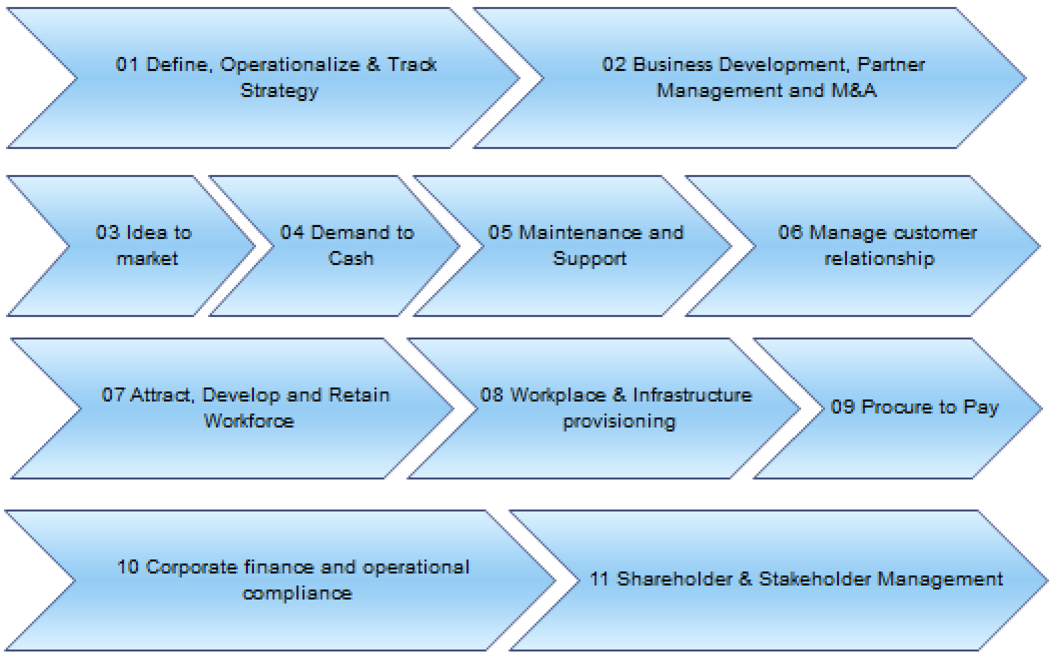
PowerDesigner lets you capture, analyze, visualize, and plan changes to your processes and capabilities and the value streams and flows you derive from them. You can create process maps to show the hierarchy of your business processes and capabilities and city planning diagrams to give a big picture view of your business capabilities and processes.

3.1 Process Maps

A process map provides a graphical view of your business architecture, and helps you identify your business capabilities and high-level processes, independent of the people and business units who fulfill them.

> **Note**
>
> With business capabilities, process groups, and value streams all supporting box-in-box display in enterprise architecture diagrams, the specialized process map diagram is deprecated and may be removed in a future release.

The following example shows a top-level process map:



To create a process map in an existing EAM, right-click the model in the Browser and select *New* > *Process Map*. To create a new model, select *File* > *New Model*, choose **Enterprise Architecture Model** as the model type and *Process Map* as the first diagram, and then click *OK*.

The following objects can be created from the process map toolbox:

Tool	Description
	Business Process - An activity performed in the enterprise. See Process Groups and Business Processes (EAM) [page 58].
	Business Capability - An aggregation of processes and/or sub-capabilities. See Business Capabilities (EAM) [page 56].
	Area - An abstract object for grouping other objects. See Areas (EAM) [page 55].
	Goal - A mission, vision, strategy, or objective. See Goals (EAM) [page 102].
	Program - A high-level EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	Project - An EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	Impact - A link from an EA initiative to the asset that it impacts. See Specifying the Impact of a Project on EA Assets [page 105].
	Fulfillment - A link from an EA initiative to a goal that it contributes to fulfilling. See Specifying the Fulfill-ment of Goals by a Project [page 104].

For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

3.1.1 Creating a Multi-Level Process Map

Process maps are commonly maintained by process analysts who, starting from a top-level overview of business areas and high-level capabilities, decompose top-level processes into sub-processes. Some or all of the processes in the top-level map are decomposed into sub-processes containing sub-maps and so on down through a number of levels. Areas and business capabilities are only permitted in the top-level and cannot be created in sub-maps.

It is common practice to decompose processes to four levels in a process map, and then to model the steps of each fourth-level process in a business process diagram (see Linking Processes to BPMN Business Process Diagrams [page 60]).

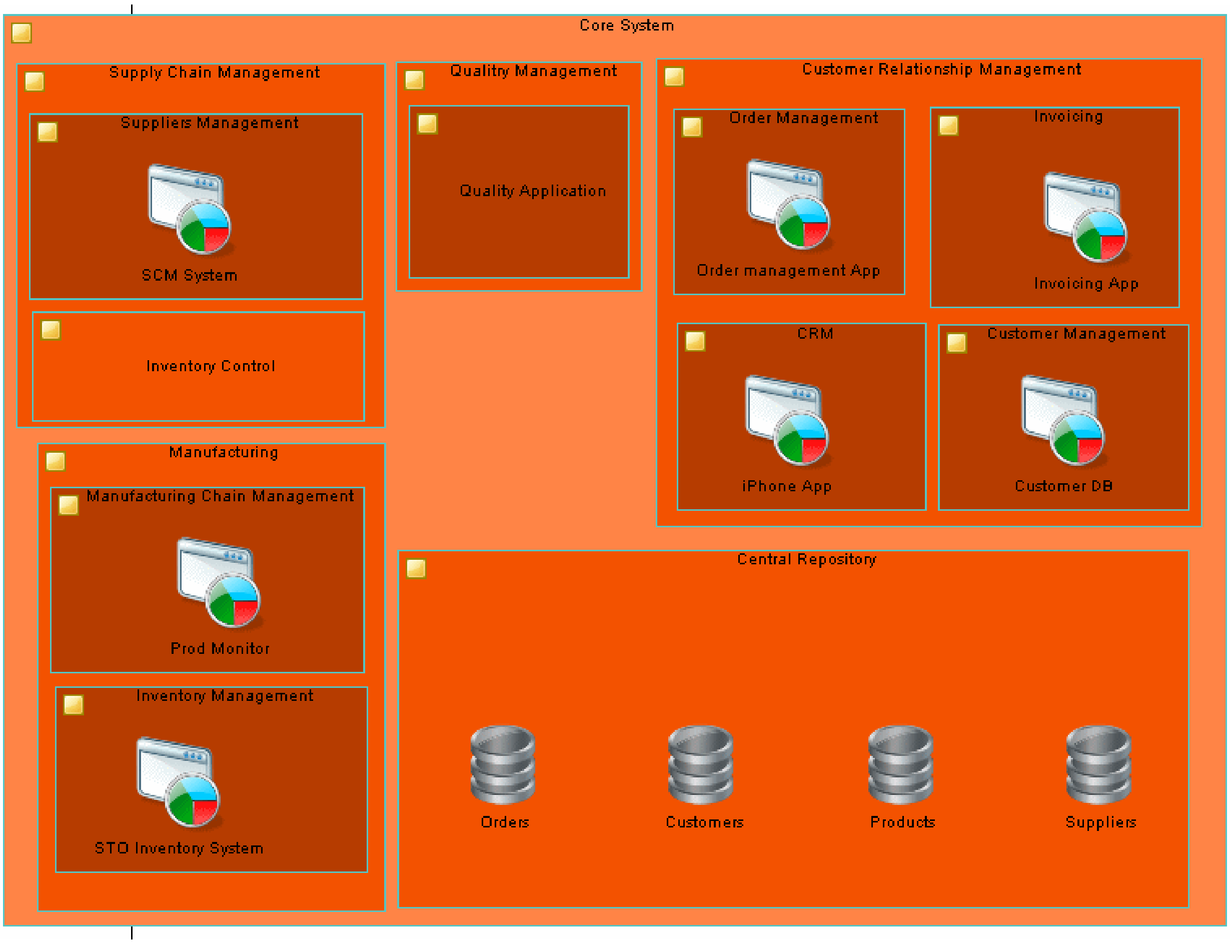
To create a sub-map inside a process (or to go down into a sub-map that has already been created, press *Ctrl* and double-click the process symbol. Processes that you create in this sub-map are sub-processes of the inital process, and are displayed under it in the Browser and on the *Processes* tab of its property sheet. To go back up to the parent map, press *Ctrl+U*.

Business capabilities can contain sub-capabilities and processes, but they cannot contain sub-maps. To create sub-capabilities or processes inside a capability, use the *Business Capability* or *Process* tool and click in an existing business capability symbol. Sub-capabilities and processes created inside a business capability are displayed under it in the Browser and on the *Sub-Capabilities* or *Processes* tab of its property sheet.

3.2 City Planning Diagrams

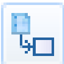
A city planning diagram provides a graphical view of the big picture of your enterprise architecture, using the metaphor of planning the infrastructure of a city to represent the organization of capabilities, systems, applications, etc into areas.

In the following example, the core system is broken up into five major sub-systems, which are in turn sub-divided into their major functions. Major applications and databases are also highlighted.



To create a city planning diagram in an existing EAM, right-click the model in the Browser and select *New* > *City Planning Diagram*. To create a new model, select *File* > *New Model*, choose Enterprise Architecture Model as the model type and *City Planning Diagram* as the first diagram, and then click *OK*.

The following objects can be created from the city planning diagram toolbox:

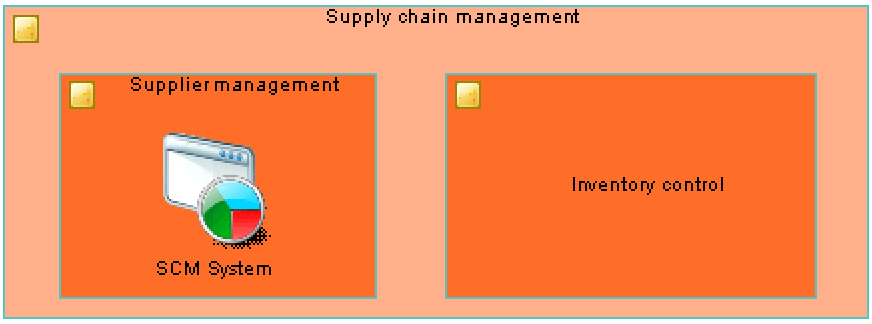
Tool	Description
	*Area* - An abstract object for grouping other objects. See Areas (EAM) [page 55].
	*System* - A packaged application. See Systems and Applications (EAM) [page 71].
	*Application* - A computer program. See Systems and Applications (EAM) [page 71].
	*Database* - A database. See Databases (EAM) [page 73].
	*Business Capability* - An aggregation of processes and/or sub-capabilities. See Business Capabilities (EAM) [page 56].
	*Program* - A high-level EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	*Project* - An EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	*Impact* - A link from an EA initiative to the asset that it impacts. See Specifying the Impact of a Project on EA Assets [page 105].

For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

3.3 Areas (EAM)

An area is an abstract object that can group together other objects. The objects do not belong to the area and are just grouped in it.

In this example, the Supply chain management area contains the Supplier management and Inventory control areas:



Creating an Area

You can create an area from the Toolbox, Browser, or *Model* menu, or on the *Sub-Areas* tab of an area.

Area Properties

To view or edit an area's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

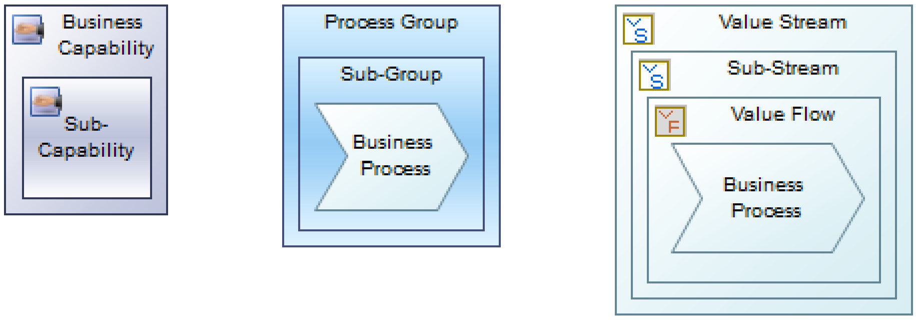
The following tabs are also available:

- Role Associations - lists the organization units, people, and roles associated with the area, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Attached Objects – lists the objects that are associated with the area.
- Sub-Areas – lists the areas contained within the area.

3.4 Business Capabilities (EAM)

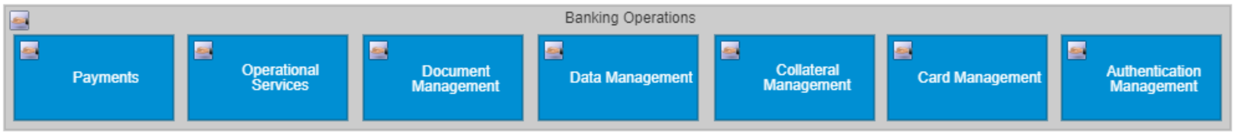
Business capabilities are an abstract representation of what a business does. Capabilities can be broken down into sub-capabilities and can be realized by business processes, which provide a concrete vision of how the business does things.

Business capabilities can contain sub-capabilities:



For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

In this example, the **`Banking Operations`** capability contains sub-capabilities:



Creating a Business Capability

You can create a business capability from the Toolbox, Browser, or *Model* menu, or on the *Sub-Capabilities* tab of a business capability.

Business Capability Properties

To view or edit a business capability's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.

Property	Description
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Level	[read-only] Specifies the level of the capability. Level 1 capabilities contain level 2 capabilities and so on.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Role Associations – lists the organization units, people, and roles associated with the capability, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Sub-Capabilities – lists the business capabilities contained within the capability.
- Business Processes – [deprecated] This list is maintained only for backwards compatibility. Processes belong to their process group and are associated with the capabilities they realize via their property sheets (see Realizing Business Capabilities via Processes or Value Flows [page 58]).
- Sites – lists the sites (see Sites (EAM) [page 39]) associated with the capability.

3.4.1 Realizing Business Capabilities via Processes or Value Flows

Business capabilities can be associated with business processes or value flows to show how they are realized.

Procedure

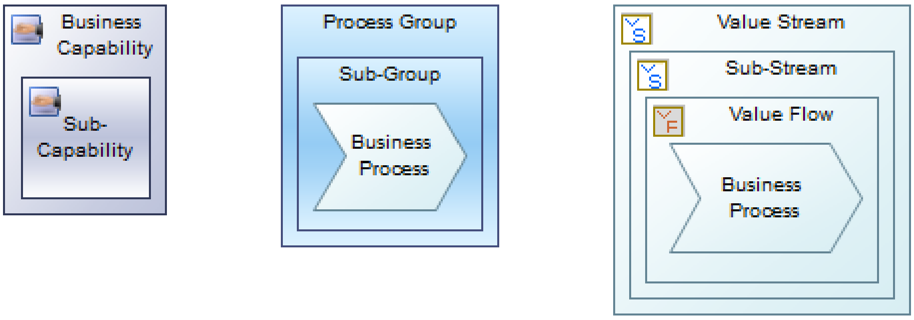
1. Open the property sheet of the business process or value flow to which you want to associate the business capability, and click the *Business Capabilities* tab.
2. Use the *Add* or *Create* tools above the list to select or create capabilities and add them to the list.

   The business processes and value flows that are associated with a capability are listed on its *Dependencies* tab in the *Realized by Processes* and *Realized by Value Flows* sub-tabs.

3.5 Process Groups and Business Processes (EAM)

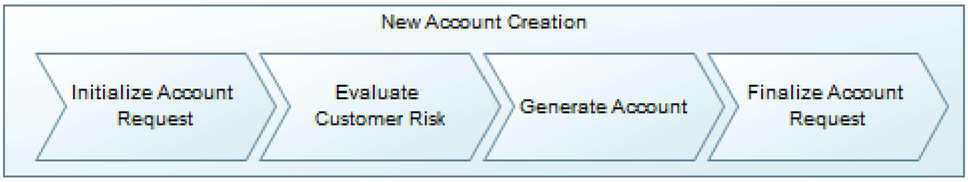
Process groups and business processes represent how things are done in the organization. Process groups can be broken down into sub-groups and the lowest level of sub-group are broken down into processes, which are in turn linked to BPMN diagrams to analyze their steps. Processes can realize business capabilities and contribute to value flows.

Process groups can contain sub-groups and business processes:



For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

In this example, the `New Account Creation` process group contains various processes:



Creating a Process Group or Business Process

You can create a process group or business process from the Toolbox, Browser, or *Model* menu, or on the appropriate tab of a process group.

Process Group and Business Process Properties

To view or edit a process group or business process's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Number ID	Specifies the number of the process in the sequence. Numbers are allotted sequentially as processes are created. If you move processes around, the numbers may no longer correspond to the desired order. You can modify the number of a process here, and the other numbers will be changed to avoid duplications and fill holes in the sequence where possible.
Reused process	[process] Specifies the name of the reused process.
Level	[process group, read-only] Specifies the level of the process group. Level 1 groups contain level 2 groups and so on.
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Role Associations – lists the organization units, people, and roles associated with the process, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Process Groups - [process groups] lists the child groups contained within the process group.
- Business Processes - lists the child processes contained within the process group.
- Business Capabilities - [business processes] lists the business capabilities realized by the process (see Realizing Business Capabilities via Processes or Value Flows [page 58]).
- Inputs / Outputs - [business processes] list the data acting as inputs and outputs to the flow (see Specifying Input and Output Data for Processes and Value Flows [page 62]).

3.5.1 Linking Processes to BPMN Business Process Diagrams

While you can decompose process groups into sub-groups and business processes, it is common practice to model the steps of these processes in a BPMN business process diagram. The process group structure is commonly maintained by an enterprise architect or process analyst, while the modeling of process steps is often done by process owners.

Context

> **Note**
>
> You must create the BPMN business process diagram before you can link it to a business process.

Procedure

1. Open the business process model containing the diagram to link to in your PowerDesigner desktop client workspace.
2. In the EAM containing your business processes, open the property sheet of the process and click the *Related Diagrams* tab.
3. Click the *Add Objects* tool and, in the dialog, select the BPM in the *Models* list.
4. Select the diagram that you want to link to and then click *OK*.

   The business process diagram is now associated with the process. You can navigate to it from the EAM process by pressing `Ctrl` and double-clicking the process symbol.

3.5.2 Associating Processes with Value Flows

You can associate business processes with value flows to show how they contribute to creating value in your enterprise. You can associate a process with a value flow by dragging it onto the value flow symbol or via the value flow's property sheet.

Procedure

1. Open the property sheet of the value flow to which you want to associate the business process, and click the *Processes* tab.
2. Use the *Add* or *Create* tools above the list to select or create processes and add them to the list.

   The value flows that are associated with a process are listed on its *Dependencies* tab in the *Contributing to Value Flows* sub-tab.

3.5.3 Specifying Input and Output Data for Processes and Value Flows

You can associate input and output data with processes and value flows.

Procedure

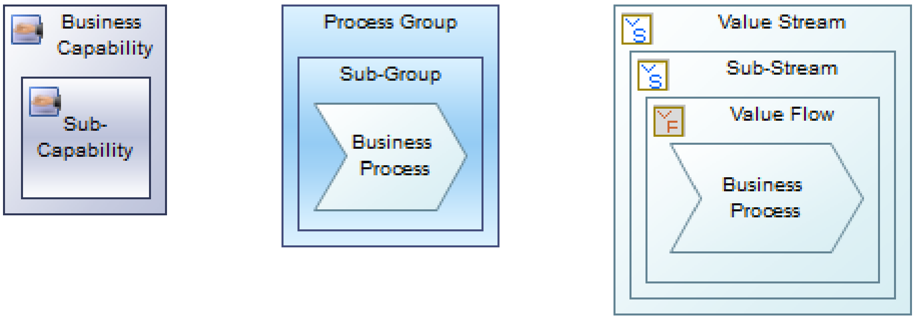
1. Open the property sheet of the process or value flow to which you want to associate the business process, and click the *Inputs* or *Outputs* tab.
2. Use the *Add* or *Create* tools above the list to select or create processes and add them to the list.

   The value flows that are associated with a process are listed on its *Dependencies* tab in the *Input to...* and *Output from...* sub-tabs.

3.6 Value Streams and Value Flows (EAM)

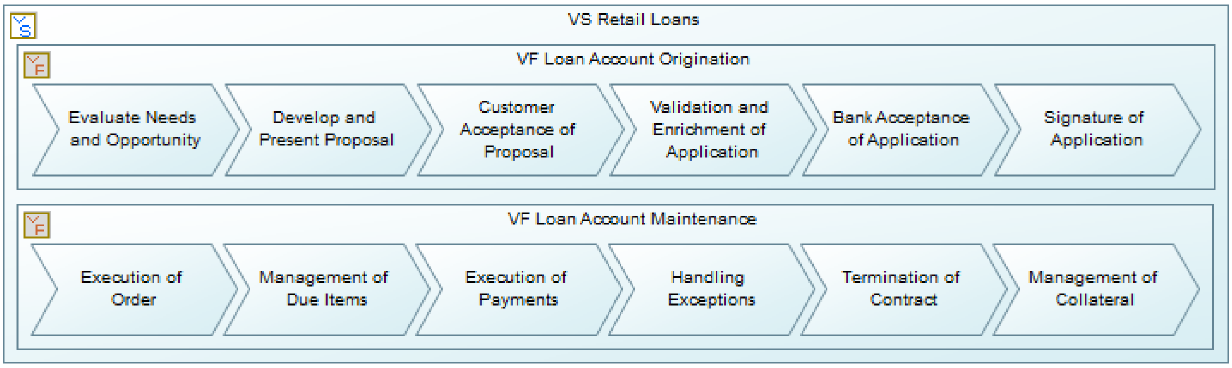
Value streams and value flows identify how value is generated by the organization. Value flows can realize business capabilities and can identify the processes that contribute to the value generation.

Value streams can contain sub-streams and value flows:



For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

In this example, the **`VS Retail Loans`** value stream contains two value flows, to which various processes contribute:



Creating a Value Stream or Value Flow

You can create a value stream or value flow from the Toolbox, Browser, or *Model* menu, or on the appropriate tab of a value stream.

Value Stream and Value Flow Properties

To view or edit a value stream or value flow's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Level	[value stream, read-only] Specifies the level of the value stream. Level 1 streams contain level 2 streams and so on.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Value Streams - [value streams] Lists the value streams contained in the stream.
- Value Flows - [value streams] Lists the value flows contained in the stream.
- Inputs / Outputs - [value flows] List the data acting as inputs and outputs to the flow (see Specifying Input and Output Data for Processes and Value Flows [page 62]).
- Business Capabilities - [value flows] Lists the business capabilities realized by the value stream or flow (see Realizing Business Capabilities via Processes or Value Flows [page 58]).
- Processes - [value flows] Lists the business processes contributing to the value generation in the flow (see Associating Processes with Value Flows [page 61]).

3.7 Importing Process Architecture Metadata from Excel

You can import your process groups, processes, business capabilities, and areas from lists in Excel files. An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File*. Select *Enterprise Architecture Model* from the model type list, enter a **`Model name`**, and then click *OK* to open the Excel Import Wizard.

   > **Note**
   >
   > You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To attach it, select *Model* > *Extensions* to open the List of Extensions, click the *Attach an Extension* tool, select **`Excel Import`** on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `02 Process Architecture Import.xlsx` contains four sheets:
   - **`BusinessCapability`** - Imports a two-level structure of business capabilities.
   - **`ProcessGroup`** - Imports a three-level structure of process groups.
   - **`Process`** - Imports business processes belonging to third-level process groups to create a typical four-level process map. The **`Parent`** column contains the code of the process group to which the process belongs. When working with a hierarchy of objects, you must use the full path to the object when

referencing it from another object. In the following example, we show the values for the **Parent**, **Code**, and **Name** columns for a hierarchy of process groups and processes:

Level	Parent	Code	Name	Full Path
1 (process group)	(none)	`P1`	`Define & Track Strategy`	`P1`
2 (process group)	`P1`	`P1_1`	`Manage Corporate Strategy`	`P1.P1_1`
3 (process group)	`P1.P1_1`	`P1_1_4`	`Manage Product Strategy`	`P1.P1_1.P1_1_4`
4 (business process)	`P1.P1_1.P1_1_4`	`P1_1_4_1`	`Manage Physical Assets`	`P1.P1_1.P1_1_4.P1_1_4_1`

- `Area` - Imports three abstract areas that can be used to loosely regroup objects into the `Management`, `Core`, and `Support` areas.

> **Note**
>
> As all the necessary metadata is contained within the file, selecting the *Auto-map columns to properties* allows the wizard to import all the objects without further intervention.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:
    - *Auto-map columns to properties* (selected)
    - All other options - deselected.
    - *Reference associated object by*: `Code`
    - *Qualified name separator*: `.` (dot)

    For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.
4. Click *Next* to begin the import.

    Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

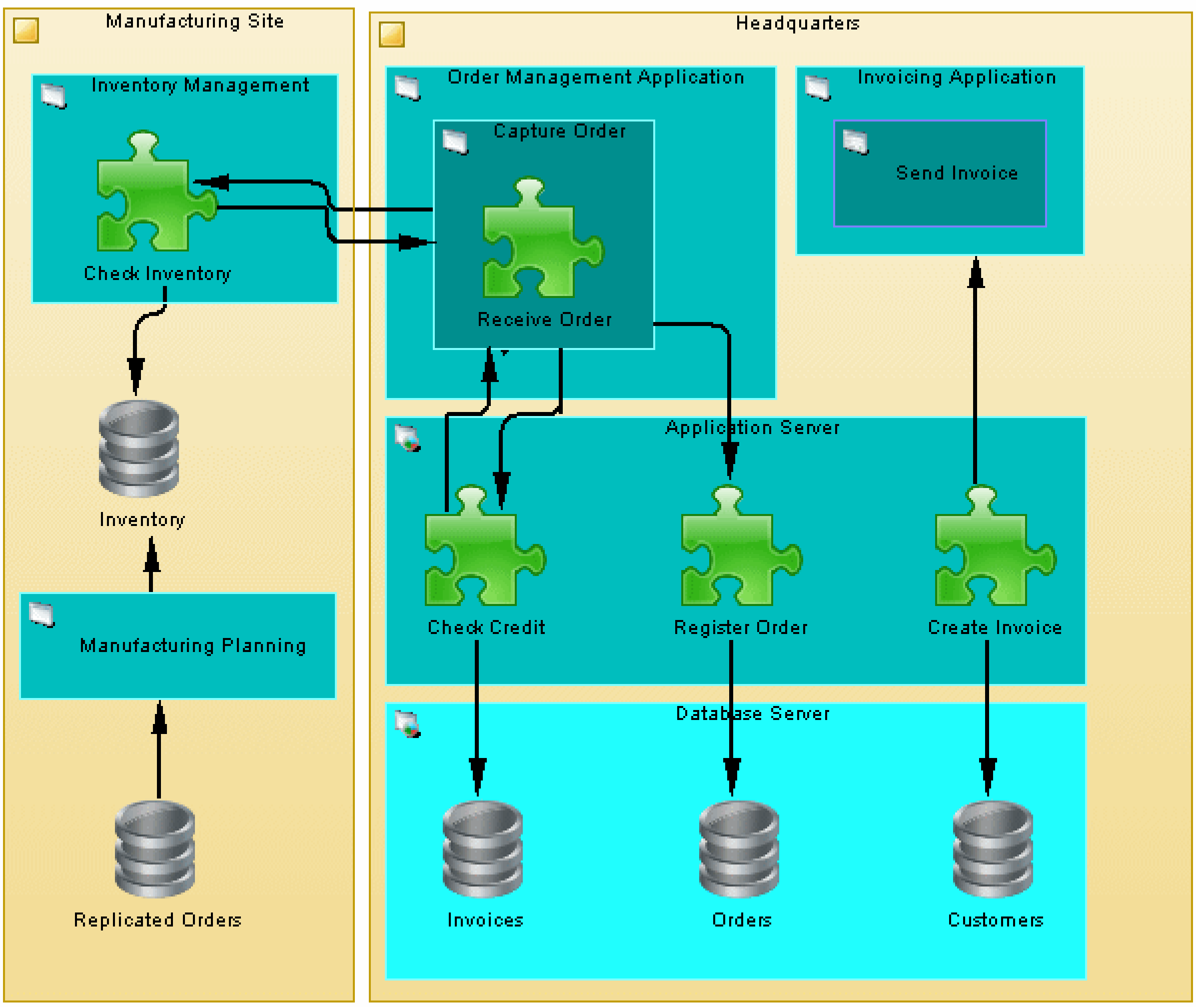
4 Application Architecture Modeling

PowerDesigner lets you capture, analyze, and visualize your systems, applications, databases, services, and other software assets. You can create application architecture diagrams to identify the structure of and interactions between your applications and components and service-oriented diagrams to organize applications and services in terms of SOA layers.

4.1 Application Architecture Diagrams

An application architecture diagram provides a high-level graphical view of the application architecture, and helps you identify applications, sub-applications, components, databases, services, etc, and their interactions.

In the following example, interactions between the major order processing and inventory systems at the company's headquarters and manufacturing site are displayed:



To create a application architecture diagram in an existing EAM, right-click the model in the Browser and select *New* > *Application Architecture Diagram*. To create a new model, select *File* > *New Model*, choose Enterprise Architecture Model as the model type and *Application Architecture Diagram* as the first diagram, and then click *OK*.

All objects can be displayed in an application architecture diagram but the primary focus is on the following software objects, which can be created from the toolbox:

Tool	Description
	Area - An abstract object for grouping other objects. See Areas (EAM) [page 55].
	Site - A physical location. See Sites (EAM) [page 39].
	Application Service - An externally visible unit of functionality. See Application and Business Services (EAM) [page 81].
	System - A packaged application. See Systems and Applications (EAM) [page 71].
	Application - A computer program. See Systems and Applications (EAM) [page 71].
	Database - A database. See Databases (EAM) [page 73].
	Component - A replaceable part of an application. See Components (EAM) [page 74].
	Form - A UI component. See Forms (EAM) [page 75].
	Document - A document. See Documents, and Reports (EAM) [page 76].
	Report Document - A report. See Documents, and Reports (EAM) [page 76].
	ETL Job - A data extraction, transformation, and load. See ETL Jobs (EAM) [page 79].
	Application Link - A link between elements in an application layer diagram. See Application Links (EAM) [page 84].
	Program - A high-level EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	Project - An EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	Impact - A link from an EA initiative to the asset that it impacts. See Specifying the Impact of a Project on EA Assets [page 105].

For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

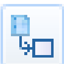
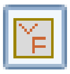
4.2 Service-Oriented Diagrams

A service-oriented diagram provides a graphical view of your business and application services and the relationships between them, and helps you associate applications and other application layer objects with business services and processes to assist with SOA design.

In the following example, the Open Account and Order-to-cash business services are shown in relation to the application services, applications, and technologies that implement them:

To create a service-oriented diagram in an existing EAM, right-click the model in the Browser and select *New* > *Service-Oriented diagram Diagram*. To create a new model, select *File* > *New Model*, choose Enterprise Architecture Model as the model type and *Service-Oriented Diagram* as the first diagram, and then click *OK*.

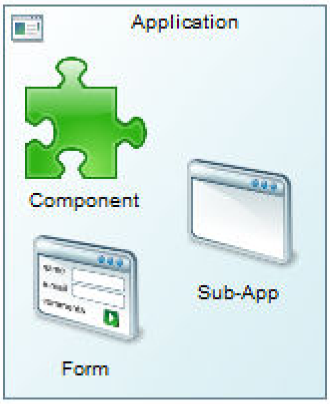
The following objects can be created from the service-oriented diagram toolbox:

Tool	Description
	*Area* - An abstract object for grouping other objects. See Areas (EAM) [page 55].
	*Role* - A set of responsibilities. See Roles (EAM) [page 47].
	*Business Process* - An activity performed in the enterprise. See Process Groups and Business Processes (EAM) [page 58].
	*Application Service* - An externally visible unit of functionality. See Application and Business Services (EAM) [page 81].
	*Business Service* - A service offered by an organization. See Application and Business Services (EAM) [page 81].
	*System* - A packaged application. See Systems and Applications (EAM) [page 71].
	*Application* - A computer program. See Systems and Applications (EAM) [page 71].
	*Database* - A database. See Databases (EAM) [page 73].
	*Component* - A replaceable part of an application. See Components (EAM) [page 74].
	*Contract* - An agreement between services. See Contracts (EAM) [page 80].
	*Application Link* - A link between elements in an application layer diagram. See Application Links (EAM) [page 84].
	*Program* - A high-level EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	*Project* - An EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	*Impact* - A link from an EA initiative to the asset that it impacts. See Specifying the Impact of a Project on EA Assets [page 105].
	*Role Association* - A link from an EA asset to a person, group, or role that is responsible for or otherwise associated with it. See Associating a Person, Organization Unit, or Role with an Object [page 44].
	
	
	

For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

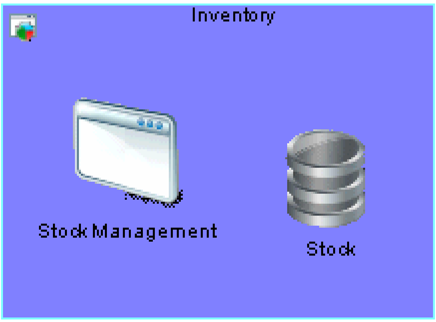
4.3 Systems and Applications (EAM)

These objects provide the primary building blocks for modeling your software assets. A system can encapsulate sub-systems, applications and applications services, databases, components, and forms. An application can encapsulate sub-applications, components, and forms.



For more information about grouping and organizing EAM objects, see Structuring EAM Objects [page 18].

In this example, the Inventory system contains the Stock Management application and the Stock database:



Creating a System or Application

You can create a system or application from the Toolbox, Browser, or *Model* menu, or on the appropriate tab of a system or application.

System and Application Properties

To view or edit a system or application's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type	[application only] Specifies the type of application. You can choose between: • Web • Rich Client • Office • Tool • CRM • ERP
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Detail – [application only] records the programming language, version, etc, for the application.
- Role Associations – lists the people, organization units, and roles associated with the application or system, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Applications – lists the applications contained within the application or system.
- Components – lists the components (see Components (EAM) [page 74]) contained within the application or system.
- Forms – lists the forms (see Documents, and Reports (EAM) [page 76]) contained within the application or system.
- Sites – lists the sites associated with the application or system (see Sites (EAM) [page 39]).
- Application Services – [system only] lists the application services (see Application and Business Services (EAM) [page 81]) contained within the system.
- Databases – [system only] lists the databases (see Databases (EAM) [page 73]) contained within the system.

- Documents / Report Documents – [system only] list the documents (see Documents, and Reports (EAM) [page 76]) contained within the system.
- ETL Jobs – [system only] lists the ETL jobs (see ETL Jobs (EAM) [page 79]) contained within the system.
- Systems – [system only] lists the sub-systems contained within the system.

4.4 Databases (EAM)

Databases store data.

Creating a Database

You can create a database from the Toolbox, Browser, or *Model* menu, or on the *Databases* tab of a system.

Database Properties

To view or edit a database's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type	Specifies the type of the database. You can choose between: • Data Warehouse • Data Mart • Multi-Dimensional Data Warehouse • OLTP Database • Virtual Database

Property	Description
DBMS / DBMS version	Specifies the DBMS of the database.
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

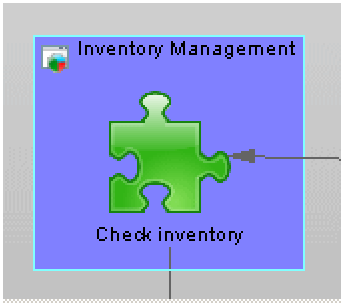
The following tabs are also available:

- Role Associations – lists the people, organization units, and roles associated with the application or system, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Source Models – lists the data models associated with the database. For more information about data models, see *Data Architecture* .
- Sites – lists the sites associated with the application or system (see Sites (EAM) [page 39]).

4.5 Components (EAM)

A component is an encapsulated, reusable, and replaceable part of an application, which can be used to implement a service or an application.

In this example, the Inventory Management system contains the Check Inventory component:



Creating a Component

You can create a component from the Toolbox, Browser, or *Model* menu.

Component Properties

To view or edit a component's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type	Specifies the type of the component. You can choose between: • EJB • Servlet • Presentation • DataAccess • Controller • COM • ActiveX
Programming Language	Specifies the programming language in which the component is written.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

4.6 Forms (EAM)

A form represents a UI component of a system or application.

Creating a Form

You can create a form from the Toolbox, Browser, or *Model* menu.

Form Properties

To view or edit a form's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type / Style	Specify the type (mobile, Web, or window) and style of the form.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Data – lists the data (see Data (EAM) [page 78]) processed by the form.

4.7 Documents, and Reports (EAM)

Documents and reports represent structured output generated and consumed in the enterprise.

In this example, the `HR` system contains a document and a report:

Creating a Document or Report

You can create a document or report from the Toolbox, Browser, or *Model* menu.

Document and Report Properties

To view or edit a document or report's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type	Specifies the type of the document or report.
Version	[document only] Specifies the version number of the document.
Category	[report only] Specifies the category of the report. You can choose between: • Business • Specification • Strategic • Technical
Period	[report only] Specifies the frequency with which the report is generated. You can choose between: • Daily • Weekly • Monthly • Yearly

Property	Description
Document Format	[document only] Specifies the XML model that is used to represent the structure of the document. Select an XSM from the list or use the tools to the right of the field to create a new XSM or view the properties of the currently selected one. For more information about XML models, see *XML Architecture* .
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Data – lists the data (see Data (EAM) [page 78]) associated with the document.

4.7.1 Data (EAM)

Data objects are pieces of information that are input or output from business processes and value flows or are used or transmitted by documents, reports, and forms. You can create a data object by selecting *Model* > *Data*.

Data Properties

To view or edit a data's properties, double-click its Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

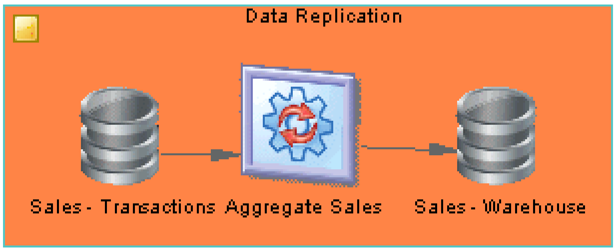
The *General* tab contains the following properties:

Property	Description
Name/Code/Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

4.8 ETL Jobs (EAM)

An ETL job represents a replication, transformation, or other movement of data.

In this example the Aggregate Sales ETL job aggregates the data from the Sales –Transactions database to the Sales – Warehouse database:



Creating an ETL Job

You can create an ETL job from the Toolbox, Browser, or *Model* menu.

ETL Job Properties

To view or edit an ETL job's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

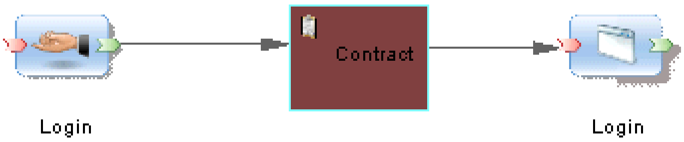
The following tabs are also available:

- Role Associations – lists the people, organization units, and roles associated with the ETL job, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).

4.9 Contracts (EAM)

A contract is an agreement between services or between an asset and a stakeholder.

In this example, the contract defines acceptable response times for the link between the business and application login services:



Creating a Contract

You can create a contract from the Toolbox, Browser, or *Model* menu.

Contract Properties

To view or edit a contract's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.

Property	Description
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Role Associations – lists the people, organization units, and roles associated with the contract, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Documents – lists the documents (see Documents, and Reports (EAM) [page 76]) associated with the contract.
- XML Models – lists the XML Models that are used to define the contract. For more information about XSMs, see *XML Architecture* .

4.10 Application and Business Services (EAM)

An application service is an externally visible unit of functionality, provided by one or more applications or components, and exposed through well-defined interfaces. A business service is a service offered by an organization to its customers that directly supports the work performed in a business process or capability, exposed by an application-to-business interface.

In this example the Register sales business service is implemented by the Finance Service application service:

Creating an Application or Business Service

You can create a business or application service from the Toolbox, Browser, or *Model* menu.

Application and Business Service Properties

To view or edit an application or business service's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type	Specifies the type of service. For business services, you can choose between: • Consulting • Customer Service • Distribution Service • Marketing Service • Personal Service • Sales Service For application services, you can choose between: • Business Process • Collaboration • Data • Infrastructure • Integration • Presentation
Quality of Service Level	Specifies the quality of the service. You can choose between: • Best effort service • Differentiated service • Guaranteed service
Security Level	Specifies the security of the service. You can choose between: • High • Moderate • Low
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Role associations – lists the organization units, people, and roles associated with the service, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Operations – lists the operations (see Business and Application Service Operations [page 83]) that support the service.

4.10.1 Business and Application Service Operations

An operation is an abstract description of an action supported by a service. For example the Login service may require a Get ID operation.

Creating an Operation

You can create operations from the property sheet of, or in the Browser under, a business or application service:

- Right-click a business or application service in the Browser, and select *New* > *Operation*.
- Click the *Add a Row* tool on the *Operations* tab of a business or application service property sheet.

Operation Properties

To view or edit an operation's properties, double-click its Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Parent	Specifies the service to which the operation belongs.
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.

Property	Description
Operation type	Specifies the type of the operation. You can choose between: • Notification • One-way • Request-Response • Solicit-Response
Input	Specifies the document or data required to start the service, which can be linked to an XML model to define its format (see Documents, and Reports (EAM) [page 76])
Output	Specifies the document or data returned by the service, which can be linked to an XML model to define its format (see Documents, and Reports (EAM) [page 76])
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

4.11 Application Links (EAM)

An application link is an oriented link that is used to connect objects in the application layer.

In this example, the Create invoice component makes a request to the Invoice database:

Creating an Application Link

You can create an application link from the Toolbox, Browser, or *Model* menu.

Application Link Properties

To view or edit an application link's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type	Specifies the type of the application link.
First object / Second object	Specify the objects at the origin and end of the link.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

4.12 Importing Application Architecture Metadata from Excel

You can import your systems, applications, and databases from lists in Excel files. An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File*. Select *Enterprise Architecture Model* from the model type list, enter a **`Model name`**, and then click *OK* to open the Excel Import Wizard.

> **Note**
>
> You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To attach it, select *Model* > *Extensions* to open the List of Extensions, click the *Attach an Extension* tool, select **`Excel Import`** on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `03 Application Architecture Import.xlsx` contains three sheets:
   - **`System`** - Imports a set of systems that are used to encapsulate applications and databases.
   - **`EnterpriseApplication`** - Imports applications belonging to a system and with values for type, version, and manufacturer.
   - **`Database`** - Imports databases belonging to a system and with values for type, DBMS, and DBMS version.

> **Note**
>
> As all the necessary metadata is contained within the file, selecting the *Auto-map columns to properties* allows the wizard to import all the objects without further intervention.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:

   - *Auto-map columns to properties* (selected)
   - All other options - deselected.
   - *Reference associated object by*: **`Code`**
   - *Qualified name separator*: **`.`** (dot)

   For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.
4. Click *Next* to begin the import.

   Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

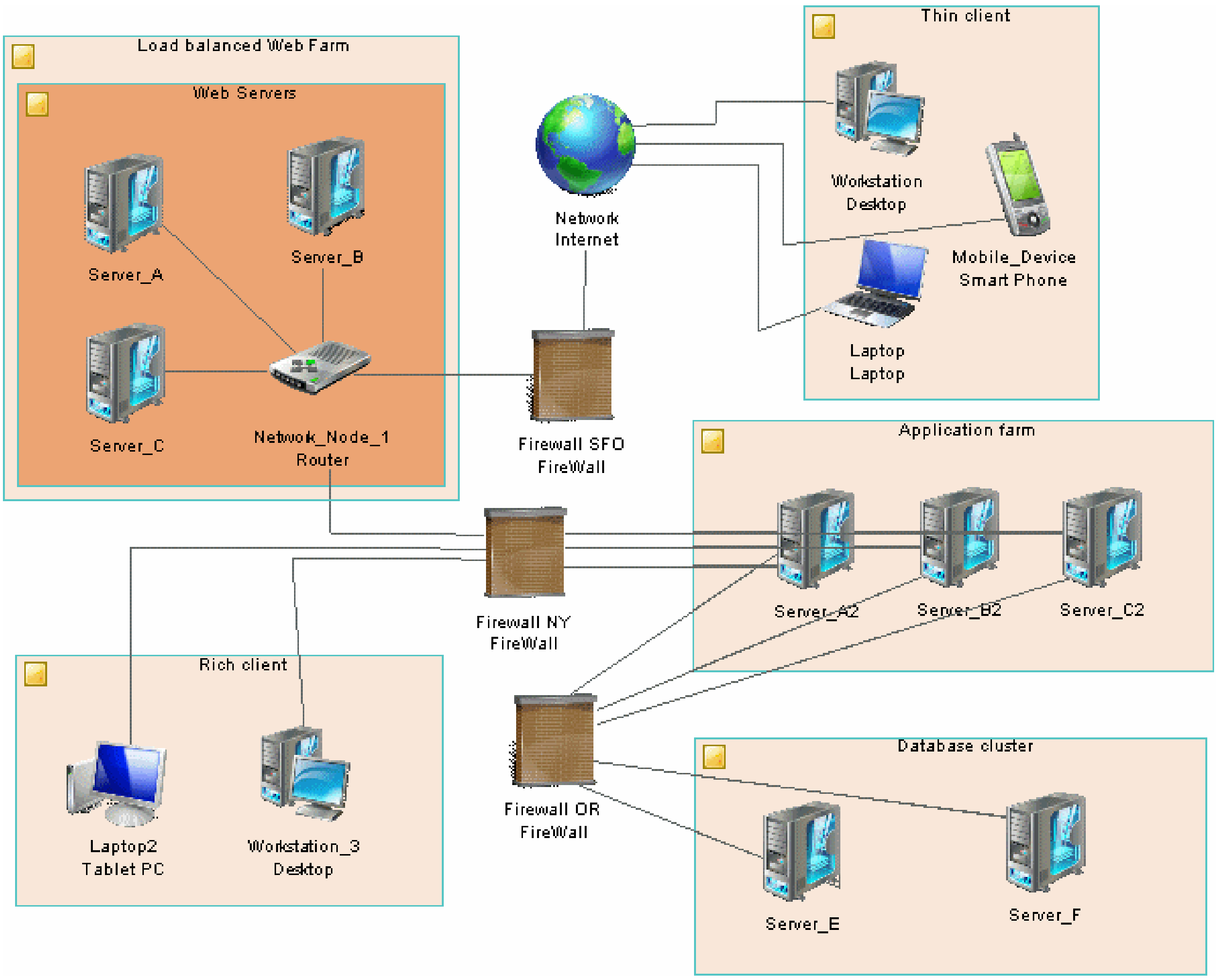
5 Infrastructure Architecture Modeling

PowerDesigner lets you capture, analyze, visualize, and plan changes to the physical infrastructure that supports your activities. You can create technology infrastructure diagrams to provide a big picture view of your networks, servers, firewalls and workstations, the applications and systems that are deployed to them and the connections between them.

5.1 Technology Infrastructure Diagrams

A technology infrastructure diagram provides a high-level graphical view of the physical architecture required to support the application architecture.

In the following example, the deployment of servers within the company network is shown, along with the web access afforded to thin clients:



To create a technology infrastructure diagram in an existing EAM, right-click the model in the Browser and select *New > Technology Infrastructure Diagram*. To create a new model, select *File > New Model*, choose Enterprise Architecture Model as the model type and *Technology Infrastructure Diagram* as the first diagram, and then click *OK*.

PowerDesigner supports all the objects necessary to build technology infrastructure diagrams:

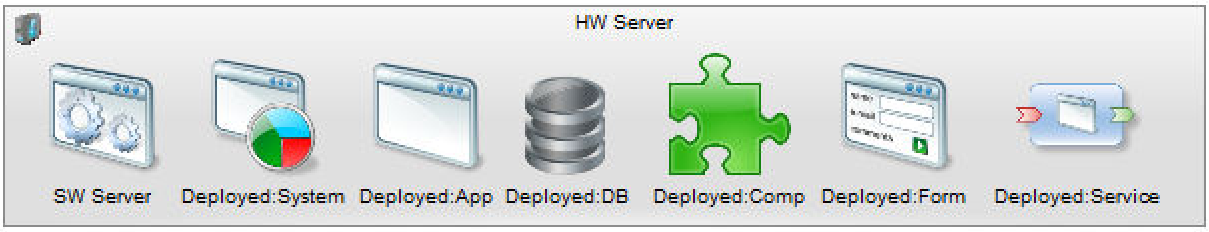
Tool	Description
	Area - An abstract object for grouping other objects. See Areas (EAM) [page 55].
	Site - A physical location. See Sites (EAM) [page 39].
	Organization Unit - A group, department, or other collection of people or organization units. See Organization Units (EAM) [page 41].
	Person - An individual. See People (EAM) [page 42].
	Role - A set of responsibilities. See Roles (EAM) [page 47].
	Network - A LAN, WAN, or other kind of network. See Networks (EAM) [page 91].
	Software Server - A commercial software environment. See Servers, Workstations, Mobile Devices, and Network Nodes (EAM) [page 89].
	Hardware Server - A high availability resource machine. See Servers, Workstations, Mobile Devices, and Network Nodes (EAM) [page 89].
	Workstation - A client machine. See Servers, Workstations, Mobile Devices, and Network Nodes (EAM) [page 89].
	Mobile Device - A handheld client. See Servers, Workstations, Mobile Devices, and Network Nodes (EAM) [page 89].
	Network Node - A hardware component connected to a network. See Servers, Workstations, Mobile Devices, and Network Nodes (EAM) [page 89].
	Deployment Instance - An instance of an object defined elsewhere. See Deployment Instances (EAM) [page 93].
	Infrastructure Link - A link between elements in a technology layer diagram. See Infrastructure Links (EAM) [page 96].
	Program - A high-level EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	Project - An EA initiative. See Programs, Projects, and Phases (EAM) [page 103].
	Impact - A link from an EA initiative to the asset that it impacts. See Specifying the Impact of a Project on EA Assets [page 105].

For more information about objects and diagrams, see Displaying EAM Objects in Diagrams [page 16].

5.2 Servers, Workstations, Mobile Devices, and Network Nodes (EAM)

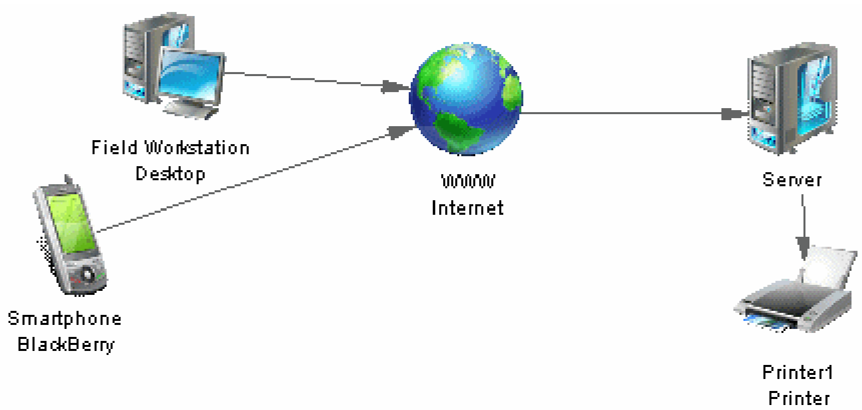
A software server is a commercial software environment, which contains a set of applications and which can be deployed to a hardware server. A hardware server is a device that serves data or other support to network devices and which usually has higher specifications than client devices. A workstation is a client machine to which an application or a server can be deployed. A mobile device is a portable client used remotely to access the network. Network nodes represent other type of network objects (routers, switches, firewalls, modems, printers, faxes, etc).

Servers, workstations, and mobile devices can contain software servers and deployed software assets:



For more information about grouping and organizing EAM objects, see Structuring EAM Objects [page 18].

In this example, the field workstation and smartphone connect via the web to the server, which is also connected to a network node of type printer:



Creating a Server, Workstation, Mobile Device, or Network Node

You can create a server, workstation, mobile device or network node from the Toolbox, Browser, or *Model* menu.

Software and Hardware Server, Workstation, Mobile Device, and Network Node Properties

To view or edit a software or hardware server, workstation, mobile device, or network node's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type	Specifies the type of the object.
Site	Specifies the site to which the object is deployed.
Multiple	[hardware server, workstation and mobile device only] Specifies that the object represents multiple machines.
Virtual	[hardware server and workstation only] Specifies that the object represents a virtual machine. This property will be selected automatically if you create or drag the object onto a hardware or software server or workstation. For more information about creating virtual machines, see Modeling Cluster Servers and Virtual Machines [page 91].
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

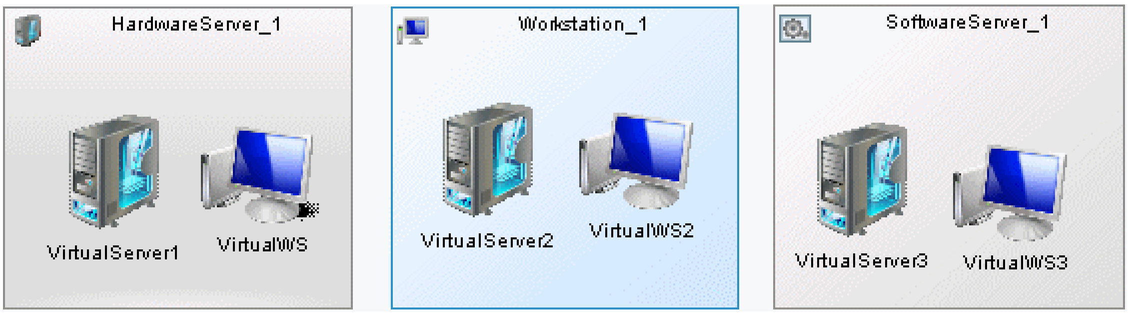
- Detail – contains information to identify the object in the environment.
- Deployment Instances – [not network node] lists the deployment instances (see Deployment Instances (EAM) [page 93]) associated with the object.
- Software Servers – [workstation, hardware server, and mobile device only] lists the software servers associated with the object.
- Role Associations – lists the people, organization units, and roles associated with the object (see Associating a Person, Organization Unit, or Role with an Object [page 44]).

5.2.1 Modeling Cluster Servers and Virtual Machines

You can model complex machine-in-machine environments using the *Type* and *Virtual* properties.

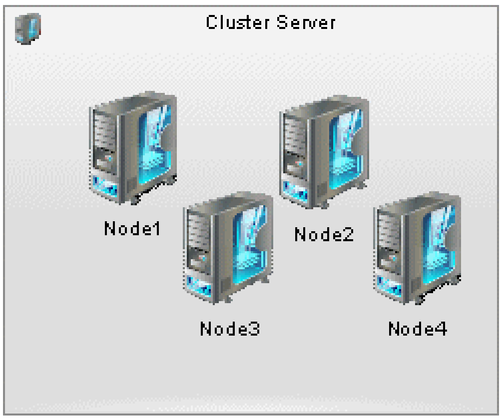
To model virtual machines, create or drag a hardware server or workstation onto a hardware or software server or a workstation. PowerDesigner will select and render read-only the *Virtual* property for objects modeled in this way. You can also model virtual machines in isolation by simply selecting the *Virtual* property yourself.

In the following example the hardware server, workstation, and software server each contain a virtual server and virtual workstation:



To model a cluster server, create a hardware server and set its *Type* to **`Cluster Server`**. You can then drop other hardware servers onto the symbol (or create them on the cluster server's *Hardware Servers* tab) to create cluster nodes. Nodes created in this way are not treated as virtual by default.

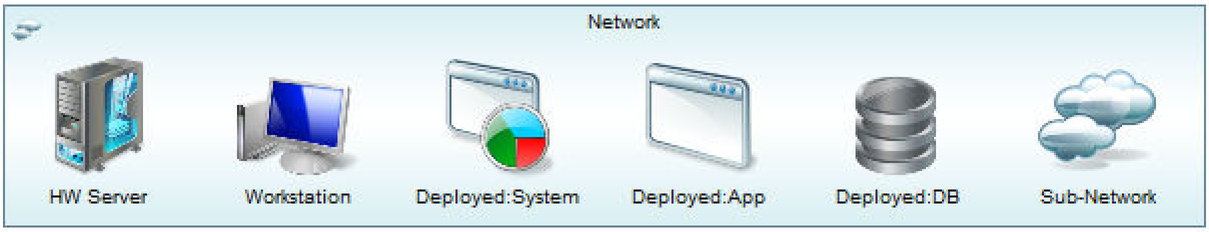
In the following example, the cluster server contains four hardware server nodes:



5.3 Networks (EAM)

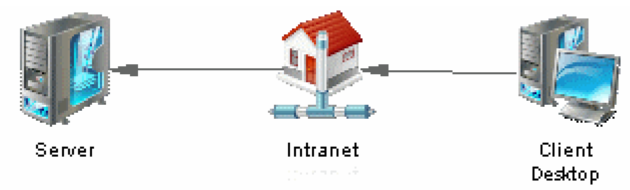
A network allows computers to communicate with each other either locally or over large distances via telecommunications.

Networks can group hardware and deployed software assets:



For more information about grouping and organizing EAM objects, see Structuring EAM Objects [page 18].

In this example, the client communicates with the server via an intranet:



Creating a Network

You can create a network from the Toolbox, Browser, or *Model* menu.

Network Properties

To view or edit a network's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.

Property	Description
Type	Specifies the type of the network.
Site	Specifies the site within which the network is located.
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- *Networks* - lists the sub-networks contained in the network.
- *Role Associations* – lists the people, organization units, and roles associated with the network (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- *Deployment Instances* – lists the deployment instances associated with the network (see Deployment Instances (EAM) [page 93]).
- *Hardware Servers*, *Software Servers*, and *Workstations* - list the servers and workstations associated with the network (see Servers, Workstations, Mobile Devices, and Network Nodes (EAM) [page 89]).

5.4 Deployment Instances (EAM)

Deployment instances represent the deployment of software objects to physical systems. You can deploy a system, application, application service, database, component, or form to multiple servers, workstations, or mobile devices. If you do not want to specify physical systems, you can deploy your software to a software server or an area, or simply to the diagram background.

Creating a Deployment Instance

To create a deployment instance of a specific software object:

- Drag a system, application, application service, database, component or form from the Browser onto a hardware or software server, workstation, mobile device, or area in a technology infrastructure diagram, or onto the diagram background.
- Open the property sheet of a hardware or software server, workstation, or mobile device, click its *Deployment Instances* tab, and use the *Add Objects* tool.

> **Note**
>
> You can, alternatively, create a deployment instance without immediately attaching it to a specific deployed object from the Toolbox, Browser, or *Model* menu, or on the *Deployment Instances* tab of a hardware or software server, workstation, or mobile device. You can, subsequently, specify the software object that the deployment

instance is deploying by opening the deployment instance property sheet and selecting the object in the *Deployed object* field.

In this example, an instance of the Suppliers database is stored on the Failover server:

Deployment Instance Properties

To view or edit a deployment instance's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Deployed Object	Specifies the system, application, application service, database, component or form being deployed. Click the buttons to the right of the field to create a new object, select an existing object, or to open the property sheet of the selected object.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

5.4.1 Importing Deployment Instances from Excel

You can import deployment instances, which are used to deploy software assets to infrastructure assets in your models, from lists in Excel files. An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File*. Select *Enterprise Architecture Model* from the model type list, enter a **Model name**, and then click *OK* to open the Excel Import Wizard.

   > **Note**
   >
   > You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To attach it, select *Model* > *Extensions* to open the List of Extensions, click the *Attach an Extension* tool, select **Excel Import** on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `06 Deployment Instances Import.xlsx` contains two sheets, deploying applications and databases to hardware servers.

   > **Note**
   >
   > Certain metadata in the file cannot be automatically evaluated by the wizard. This file cannot be imported using the *Auto-map columns to properties* option and you must specify the mappings in the wizard.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:
   - All options - deselected.
   - *Reference associated object by*: **Code**
   - *Qualified name separator*: **.** (dot)

   For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.
4. Click *Next* to begin the import.

   Specify how your sheets and columns will be imported in the wizard as follows:
   - **Application Deployments** - Creates deployment instances, linking an application to a hardware server:

1. Import the table as **`Deployment Instance`**.
2. Accept the defaults for importing the columns **`Parent`**, **`Name`**, and **`Code`**.
3. Import the column **`DeployedObject`** as **`Deployed Object`** and set the *Object* type to **`Application`**.

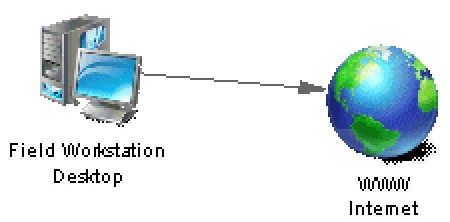
- **`Database Deployments`** - Creates deployment instances, linking a database to a hardware server:
  1. Import the table as **`Deployment Instance`**.
  2. Accept the defaults for importing the columns **`Parent`**, **`Name`**, and **`Code`**.
  3. Import the column **`DeployedObject`** as **`Deployed Object`** and set the *Object* type to **`Database`**.

Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

5.5 Infrastructure Links (EAM)

An infrastructure link is an oriented link that is used to connect objects in the technology layer.

In this example, the Field Workstation accesses the internet:



Creating an Infrastructure Link

You can create an infrastructure link from the Toolbox, Browser, or *Model* menu.

Infrastructure Link Properties

To view or edit an infrastructure link's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Type	Specifies the type of the infrastructure link.
First /second object	Specify the objects at the origin and end of the link.
Protocol	Specifies the protocol of the infrastructure link.
Physical type	Specifies the physical type of the infrastructure link.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

5.6 Importing Infrastructure Architecture Metadata from Excel

An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File*. Select *Enterprise Architecture Model* from the model type list, enter a **`Model name`**, and then click *OK* to open the Excel Import Wizard.

   > **Note**
   >
   > You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To

attach it, select **Model > Extensions** to open the List of Extensions, click the *Attach an Extension* tool, select **`Excel Import`** on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `04 Infrastructure Architecture Import.xlsx` contains one sheet:
   - **`HardwareServer`** - Imports a set of servers deployed to sites and with values for operating system and version, memory, and mass storage.

   Note

   As all the necessary metadata is contained within the file, selecting the *Auto-map columns to properties* allows the wizard to import all the objects without further intervention.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:
   - *Auto-map columns to properties* (selected)
   - All other options - deselected.
   - *Reference associated object by*: **`Code`**
   - *Qualified name separator*: **`.`** (dot)

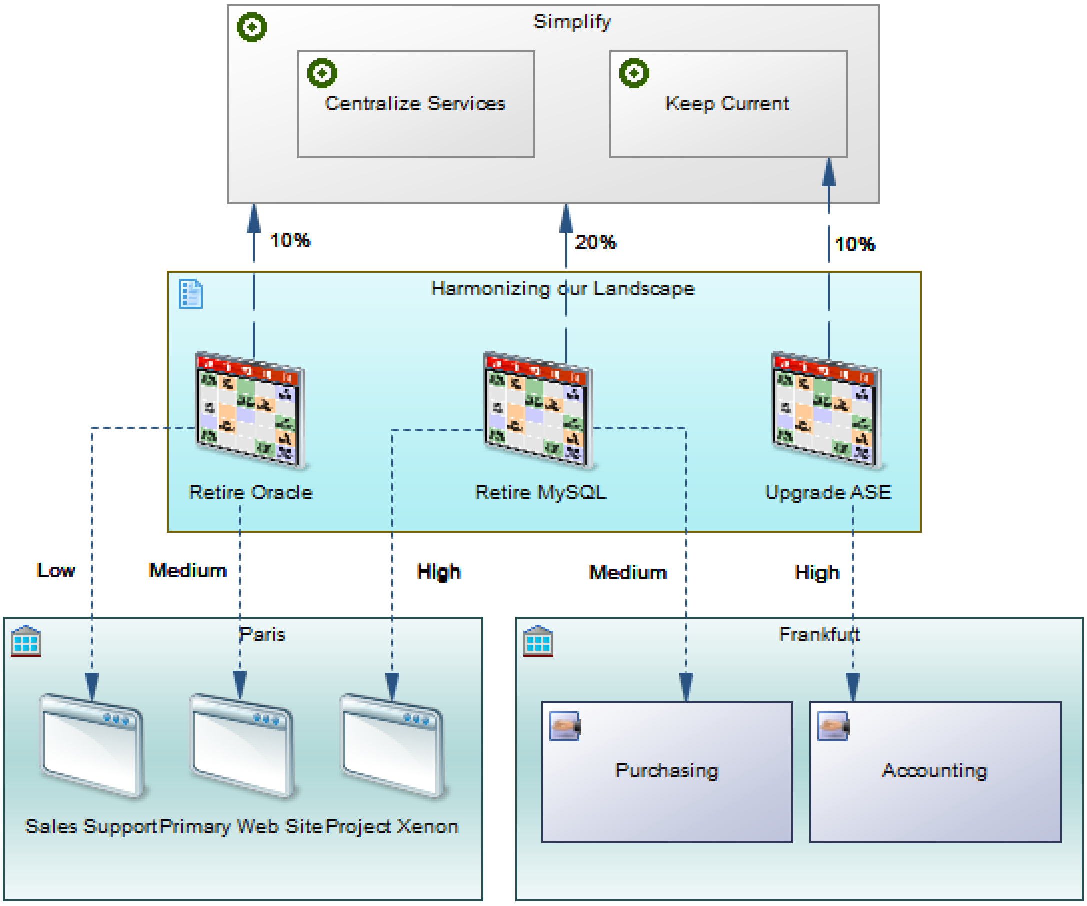
   For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.
4. Click *Next* to begin the import.

   Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

6 Goal and EA Project Modeling

PowerDesigner lets you visualize your organization's goals, how your EA projects will help fulfill them, and the impacts those projects will have on your enterprise assets.

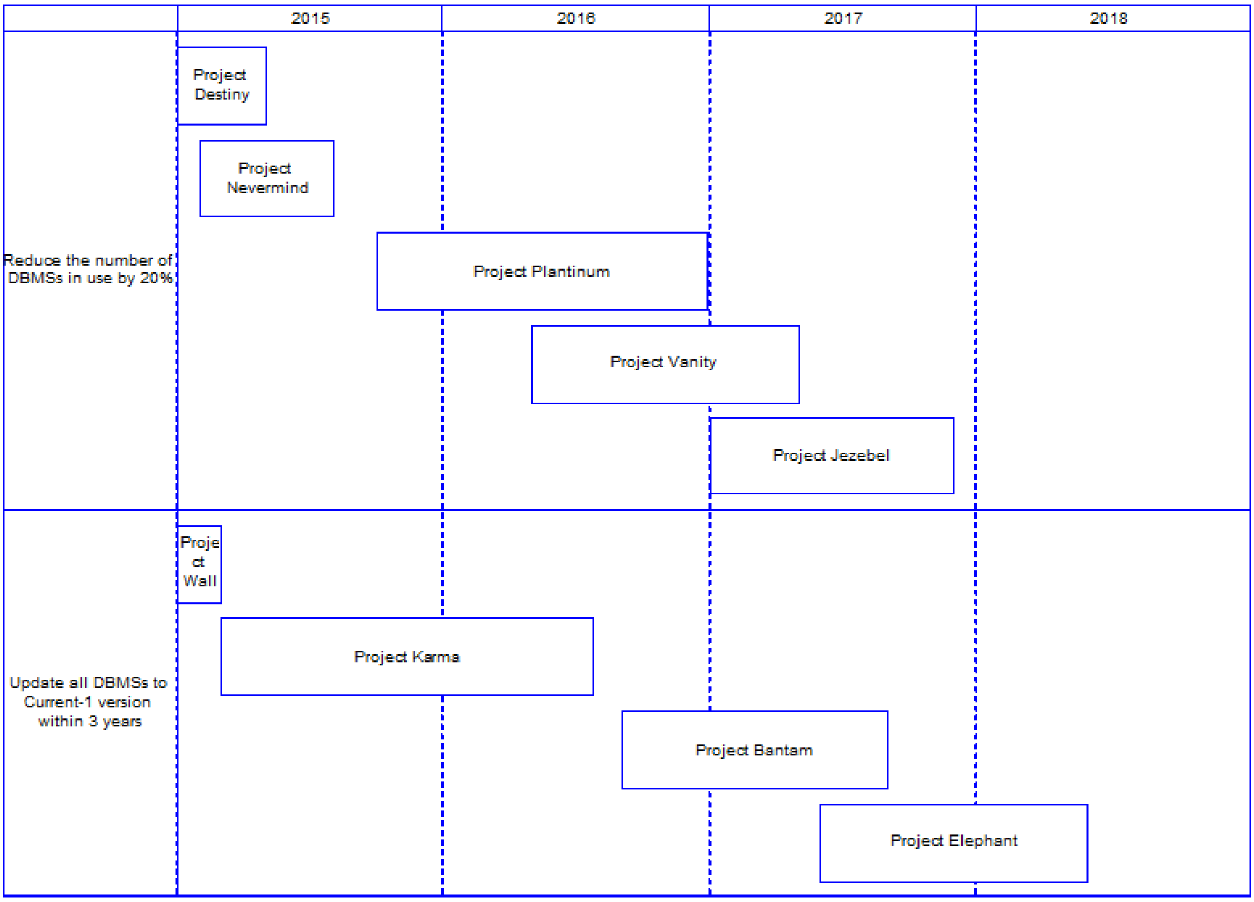
In this example, the Simplify goal contains two sub-goals and we show how the three projects within the Harmonizing our Landscape program help to realize these goals and the impacts that they will have on a number of applications and capabilities:



6.1 Timeline Diagrams

A timeline diagram provides a calendar-style overview of your enterprise architecture programs and projects in relation to your goals, or to the assets that they impact or are in other ways related to.

In the following example, the various projects intended to fulfill two selected goals are displayed in the timeline:



To create a timeline diagram in an existing EAM, right-click the model in the Browser and select *New* > *Timeline Diagram*, and then complete the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Period	Specifies the start and end dates that define the limits of the timeline, along with the time units that can be displayed in the header.
Initiative type	Specifies whether projects or programs are displayed on the timeline. If you want to show projects or programs that are not available directly under the model (for example, projects contained in programs) or if you want to specify grouping by another object, you must click the *Advanced* button and specify the path by which to obtain the projects starting from the model root and passing, by any grouping object. In the example above, goals are modeled in three levels, with third-level goals linked to projects. The type is set to `Project` [path: `Goals` (`Goal`) > `SubGoals` (`Goal`) > `SubGoals` (`Goal`) > `AssociatedProjects` (`Project`)]

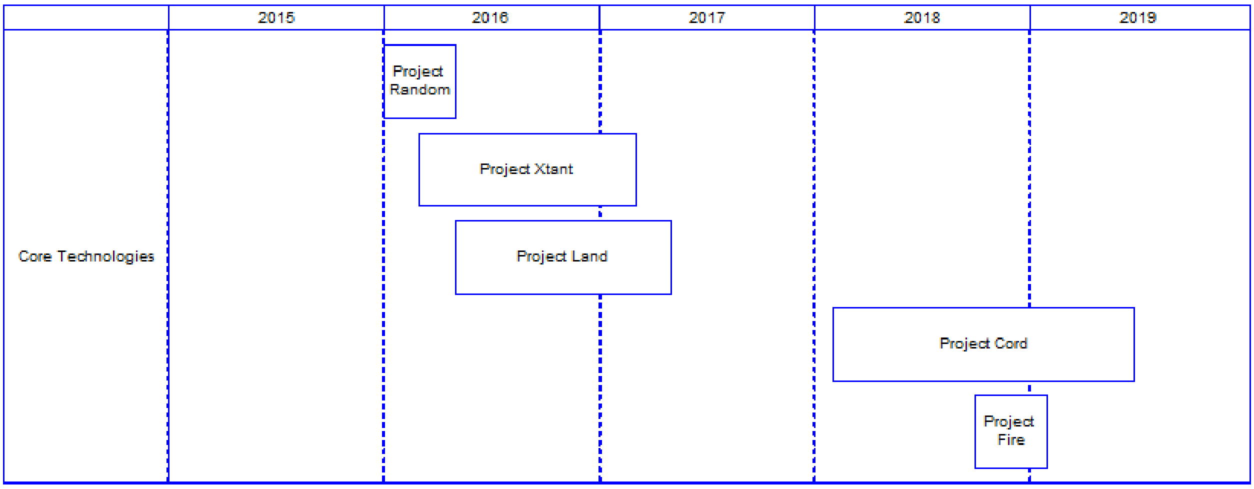
Property	Description
Group by	Specifies the objects that are displayed in the left column of the timeline and provide lanes in which the projects or programs are displayed. Click the *Selection* button to limit the number of lanes displayed. In the example above, grouping is by goals, and two specific goals are selected.

> **Note**
>
> Timeline diagrams are not editable; they are calculated based on the parameters you define in their property sheets. To reopen the property sheet at any time click in the timeline header or in empty space in any of the lanes.
>
> If changes you make to objects displayed in the timeline are not reflected in the timeline, select *View* > *Redisplay* (or press `Shift` + `F5`) to refresh it.

You can modify the linestyle of the timeline and the look and contents of the project or program symbols by right-clicking the diagram background outside of the timeline and selecting *Display Preferences*.

Example: Projects and Programs

In this example, projects are displayed in relation to the Core Technologies sub-program to which they belong:

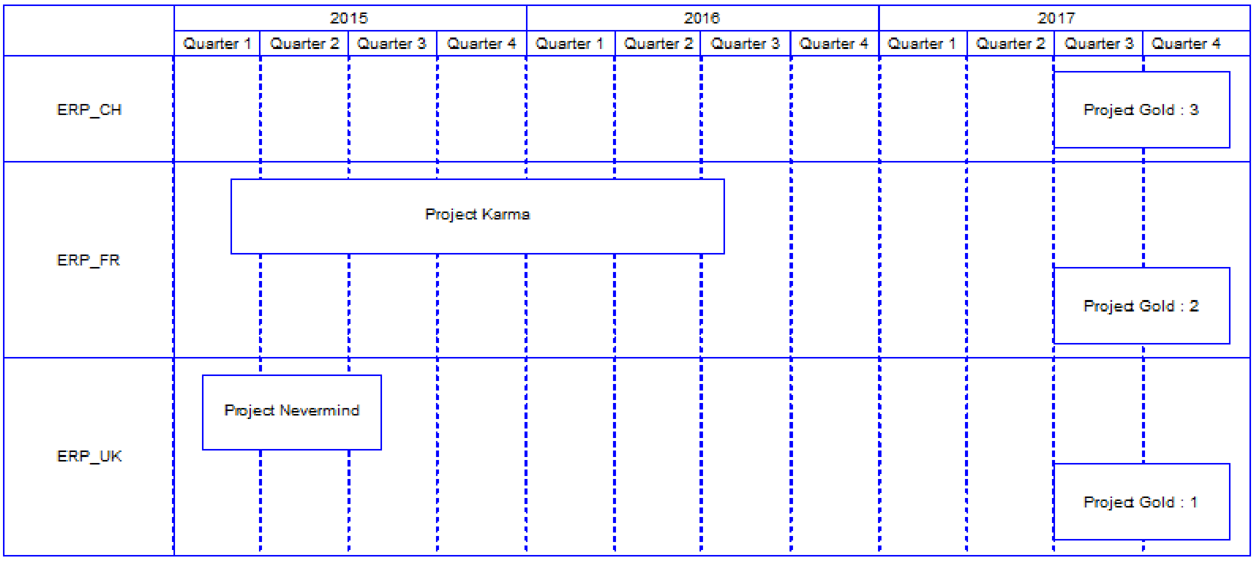


In the model, programs contain sub-programs, which contain projects. The diagram is defined as follows:

- *Initiative type*: **`Project`** [path: **`Programs`** (**`Program`**) > **`SubPrograms`** (**`Program`**) > **`Projects`** (**`Project`**)]
- *Group by*: **`Program`** (one program selected)

Projects and Systems

In this example, projects are displayed in relation to the ERP systems they impact:



In the model, systems are linked to projects by impact links. The diagram is defined as follows:

- *Initiative type*: `Project` [path: `Systems` (`System`) > `ImpactingProjects` (`Project`)]
- *Group by*: `System` (three systems selected)

6.2 Goals (EAM)

Goals allow you to model your organization's mission, vision, strategy, and objectives, to show how they are related to your organization's business and IT architecture, and how they will be addressed through enterprise architecture initiatives.

Creating a Goal

You can create a goal from the Toolbox, Browser, or *Model* menu, or from the *Sub-Goals* tab of a goal.

Goal Properties

To view or edit a goal's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Role Associations – lists the organization units, people, and roles associated with the goal, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Sub-Goals – lists the child goals contained within the goal.
- Projects – lists the contributions of EA programs and projects towards fulfilling the goal (see Specifying the Fulfillment of Goals by a Project [page 104]).

6.3 Programs, Projects, and Phases (EAM)

Programs and projects represent EA initiatives that are intended to fulfill corporate goals and may impact elements of your organization's business and IT architecture. Programs can contain sub-programs and projects, and projects can contain phases.

Creating a Program, Project, or Phase

You can create a program or project from the Toolbox, Browser, or *Model* menu. Projects can be created on the *Projects* tab of a program, and phases on the *Phases* tab of a project.

Program, Process, and Phase Properties

To view or edit these objects' properties, double-click their diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Stereotype	Extends the semantics of the object. You can enter a stereotype directly in this field, or add stereotypes to the list by specifying them in an extension file.
Start date/End date	Specify the duration of the program, project, or phase.
Progress	Specifies the current advancement of the program, project, or phase.
Status	Specifies the present status of the program, project, or phase.
IT Capital Expenditure/IT Operating Expenditure/Business Expenditure	Specify values for each of these types of expenditure for the program, project, or phase.
Owner	Specifies the person, role, or organization unit that is responsible for the object. Select an owner directly here, or create a role association of type `Owner`.
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

The following tabs are also available:

- Role Associations – lists the organization units, people, and roles associated with the program, project, or phase, and the type of role they play in relation to it (see Associating a Person, Organization Unit, or Role with an Object [page 44]).
- Sub-Programs - [programs] lists the child programs under the program. Use the *Add a Row* tool on this tab to create a new program.
- Projects - [programs] lists the projects under the program. Use the *Add a Row* tool on this tab to create a new project.
- Impacts - lists the EA assets that are impacted by the program, project, or phase (see Specifying the Impact of a Project on EA Assets [page 105]).
- Associated Goals - lists the goals that the program, project, or phase contributes to fulfilling (see Specifying the Fulfillment of Goals by a Project [page 104]).

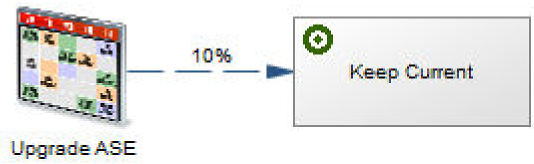
6.3.1 Specifying the Fulfillment of Goals by a Project

You can specify the fulfillment of goals by projects (or programs or phases) using the Toolbox *Fulfillment* tool or on the *Goals* tab of the property sheet of the project. The fulfillment link is displayed in the diagram, and the list of projects contributing to the fulfillment of a goal is available from the *Dependencies* tab of the property sheet of the goal.

Context

To create the fulfillment directly in a business communication diagram or process map, click the *Fulfillment* tool in the Toolbox, and draw a link from the project (or program or phase) to the goal. Double-click the link to open its property sheet and specify a fulfillment percentage and description.

In this example the Upgrade ASE project fulfills 10% of the Keep Current goal:



To specify fulfillment of goals in any diagram, including those where the *Fulfillment* tool is not available:

Procedure

1. Open the property sheet of the project (or program or phase) that is contributing to the fulfillment of a goal and select the *Goals* tab.
2. Click the *Add Objects* tool to open a dialog listing the available goals, select the goals that the project will realize, and click *OK* to return to the *Goals* tab.
3. [optional] Specify a fulfillment percentage and description to explain how the project will fulfill the goal.
4. Click *OK* to return to the diagram.
   If the project and goal are in the current diagram then the link will be drawn between them.

> **Note**
>
> To display programs, projects, and goals in a diagram drag them from the Browser, or select *Symbol* > *Show Symbols* and choose the relevant objects from the dialog. To display fulfillment, impact, and other links in the diagram, select *Tools* > *Complete Links*.

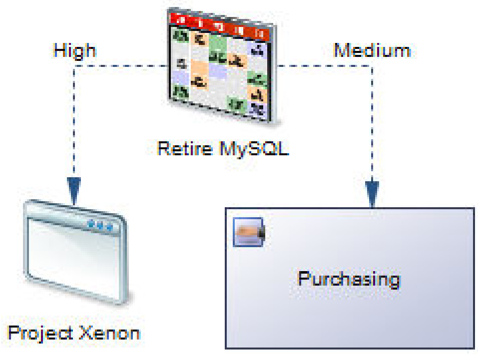
6.3.2 Specifying the Impact of a Project on EA Assets

You can specify the impact of projects (or programs or phases) on your enterprise assets (any object in the EAM) using the Toolbox *Impact* tool or on the *Impacts* tab of the project property sheet. The impact link is displayed in the diagram, and the list of projects impacting an asset is available from the *Dependencies* tab of the property sheet of the asset.

Context

To create the impact directly in the diagram, click the *Impact* tool in the Toolbox, and draw a link from the project (or program or phase) to the asset. Double-click the link to open its property sheet and specify an impact level and description.

In this example the Retire MySQL project has a high impact on the Project Xenon application and a medium impact on the Purchasing business capability:



To specify impacts on assets from the property sheet of a project, program, or phase:

Procedure

1. Open the property sheet of the project (or program or phase) that is impacting an EA asset and select the *Impacts* tab.
2. Click the *Add Objects* tool to open a dialog listing the available assets, select those that the project will impact, and click *OK* to return to the *Impacts* tab.
3. [optional] Specify an impact level and description to explain how the project will impact the asset.
4. Click *OK* to return to the diagram.

   If the project and asset are in the current diagram then the link will be drawn between them.

   > **Note**
   >
   > To display programs, projects, and goals in a diagram drag them from the Browser, or select *Symbol* > *Show Symbols* and choose the relevant objects from the dialog. To display fulfillment, impact, and other links in the diagram, select *Tools* > *Complete Links*.

6.4 Analysis Criteria (EAM)

Analysis criteria allow you to define a benchmark, standard, or scale, and then score your objects against it.

For example, you could set the following criteria:

- `Cloud Readiness` - Rank your business capabilities on a scale of `1-5`.
- `24/7 Availability` - Assign a scale of importance `Low, Medium, High` for each of your systems.

Criteria can be used to drive heat map coloring in PowerDesigner Web.

Creating a Criterion

You can create a criterion by selecting *Model* > *Analysis Criteria* and clicking the *Add a Row* tool.

Criterion Properties

To view or edit a criterion's properties, double-click its Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.

Property	Description

Category/Type	Specifies the category/type of criteria. You can choose from the following standard categories/types or enter your own:
	• `Maturity`:
	  ○ `Architecture`
	  ○ `Change`
	  ○ `Performance`
	  ○ `Result`
	• `Transformation`:
	  ○ `Risk`
	  ○ `Objective`
	  ○ `Issue`
	  ○ `Scope`
	• `Impact`:
	  ○ `Financial`
	  ○ `Regulatory`
	  ○ `Reputation`
	  ○ `Customer`
	• `<no category>`
	  ○ `Pain Point`
	  ○ `Control`
	  ○ `Goal`
	  ○ `Scope`

List of Values	Specifies a list of values available to assign to objects for the criterion. You can choose from the following standard value lists or enter your own comma-separated list: • `1,2,3,4,5` • `Low,Medium,High` • `Very Low,Low,Medium,High,Very High`
Keywords	Provide a way of loosely grouping objects through tagging. To enter multiple keywords, separate them with commas.

6.4.1 Setting Criteria Values for EA Assets

You can set criteria values for your enterprise assets via the asset property sheet *Analysis Criteria* tab.

Procedure

1. Open the property sheet of the object and click the *Analysis Criteria* tab.
2. Click the *Add a Row* tool, and select a criterion from the list.
3. Select a value from among the permitted values for the criterion in the *Values* list.

6.5 Importing Corporate Goals and EA Projects from Excel

You can import goals and the programs and projects that will seek to fulfill them from lists in Excel files. An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File* . Select *Enterprise Architecture Model* from the model type list, enter a **`Model name`**, and then click *OK* to open the Excel Import Wizard.

   > **Note**
   >
   > You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To attach it, select *Model* > *Extensions* to open the List of Extensions, click the *Attach an Extension* tool, select **`Excel Import`** on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `07 Goals and Projects Import.xlsx` contains three sheets:
   - **`Goal`** - Imports a three-level set of corporate goals.
   - **`Program`** - Imports a two-level set of programs that will contain projects for fulfilling the goals.

- `Project` - Imports a set of projects that belong to second-level programs.

> Note
>
> As all the necessary metadata is contained within the file, selecting the *Auto-map columns to properties* allows the wizard to import all the objects without further intervention.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:
   - *Auto-map columns to properties* (selected)
   - All other options - deselected.
   - *Reference associated object by*: `Code`
   - *Qualified name separator*: `.` (dot)

   For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.
4. Click *Next* to begin the import.

   Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

6.5.1 Importing Impacts, Fulfillments, and Stakeholders from Excel

You can import links from projects to other objects in your models from lists in Excel files. An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File*. Select *Enterprise Architecture Model* from the model type list, enter a `Model name`, and then click *OK* to open the Excel Import Wizard.

   > Note
   >
   > You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To attach it, select *Model* > *Extensions* to open the List of Extensions, click the *Attach an Extension* tool, select `Excel Import` on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `08 Impacts Fulfillments and Stakeholders Import.xlsx` contains four sheets linking programs and projects to various other objects created in precedent imports.

   > **Note**
   >
   > Certain metadata in the file cannot be automatically evaluated by the wizard. This file cannot be imported using the *Auto-map columns to properties* option and you must specify the mappings in the wizard.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:
   - All options - deselected.
   - *Reference associated object by*: **`Code`**
   - *Qualified name separator*: **`.`** (dot)

   For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.

4. Click *Next* to begin the import.

   Specify how your sheets and columns will be imported in the wizard as follows:
   - **`Program.Sponsors`** - Specifies people as sponsors of programs:
     1. Import the table as **`Program.Role Association`**.
     2. Import the column **`Parent`** as **`Parent`**.
     3. Import the column **`Role Player`** as **`Role Player`** and set the *Object* type to **`Person`**.
     4. Import the column **`Type`** as **`Type`**.
   - **`Project.Sponsors`** - Specifies people as sponsors of projects:
     1. Import the table as **`Project.Role Association`**.
     2. Import the column **`Parent`** as **`Parent`**.
     3. Import the column **`Role Player`** as **`Role Player`** and set the *Object* type to **`Person`**.
     4. Import the column **`Type`** as **`Type`**.
   - **`Project.Impact`** - Links projects to the systems that they will impact:
     1. Import the table as **`Project.Impact`**.
     2. Import the column **`Parent`** as **`Parent`**.
     3. Import the column **`EnterpriseAsset`** as **`Enterprise Asset`** and set the *Object* type to **`System`**.
     4. Import the column **`ImpactLevel`** as **`Impact Level`**.
   - **`Project.Fulfillment`** - Links projects to the goals that they will fulfill:
     1. Import the table as **`Project.Fulfillment`**.
     2. Import the column **`Parent`** as **`Parent`**.
     3. Import the column **`Goal`** as **`Goal`**.
     4. Import the column **`Fulfillment`** as **`Fulfillment`**.

   Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

7 SAP IQ Reference Architecture Model

PowerDesigner provides a special EAM model to help you analyze the architecture required to deploy a SAP® IQ data warehouse solution suitable for your workload. An advisor wizard generates architectures based on one or more hardware servers, and comparison tools help you choose the best architecture based on your requirements for cost and speed.

7.1 Creating a Reference Architecture Model

PowerDesigner provides a model template from which to create a reference architecture model, which by default is available in the *Technology* category of the *New Model* dialog.

Procedure

1. Select *File* > *New Model*, and click the *Categories* button.
2. Select the *Technology* category and then select the *SAP IQ Reference Architecture* template.

   > **Note**
   >
   > If your *New Model* dialog is configured in such a way that this category or template are not available, choose *Model Types* > *Enterprise Architecture Model* > *Technology Infrastructure Diagram* and then click the *Select Extensions* button and attach the `SAP IQ Reference Architecture` extension.

3. Enter a name for your model and click *OK* to create and open it.

7.2 Creating Reference Architectures

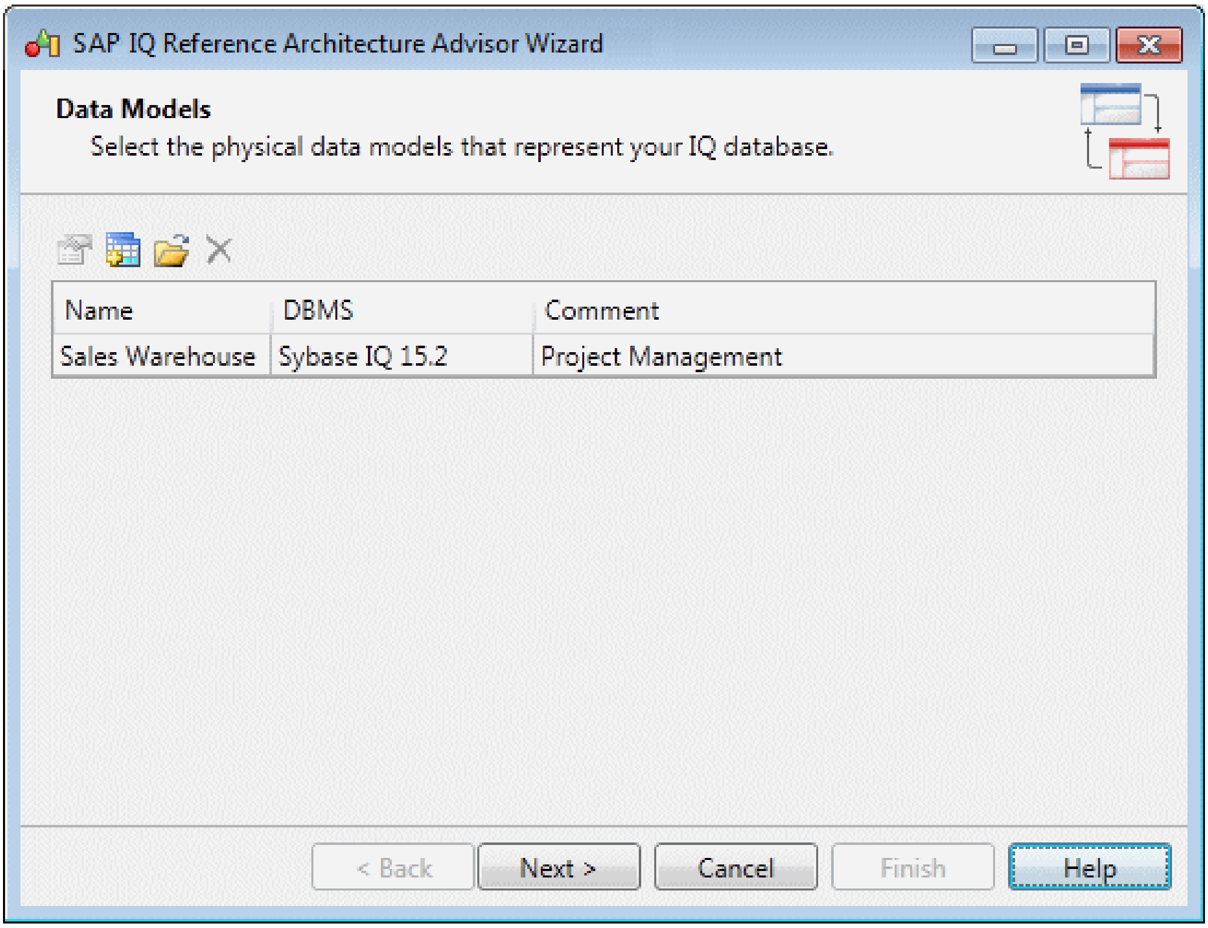
PowerDesigner provides a wizard to let you generate one or more reference architectures appropriate for your IQ environment, each based on a different hardware profile. The resulting diagram and objects detail the costs and specifications for all the elements required for each reference architecture.

Context

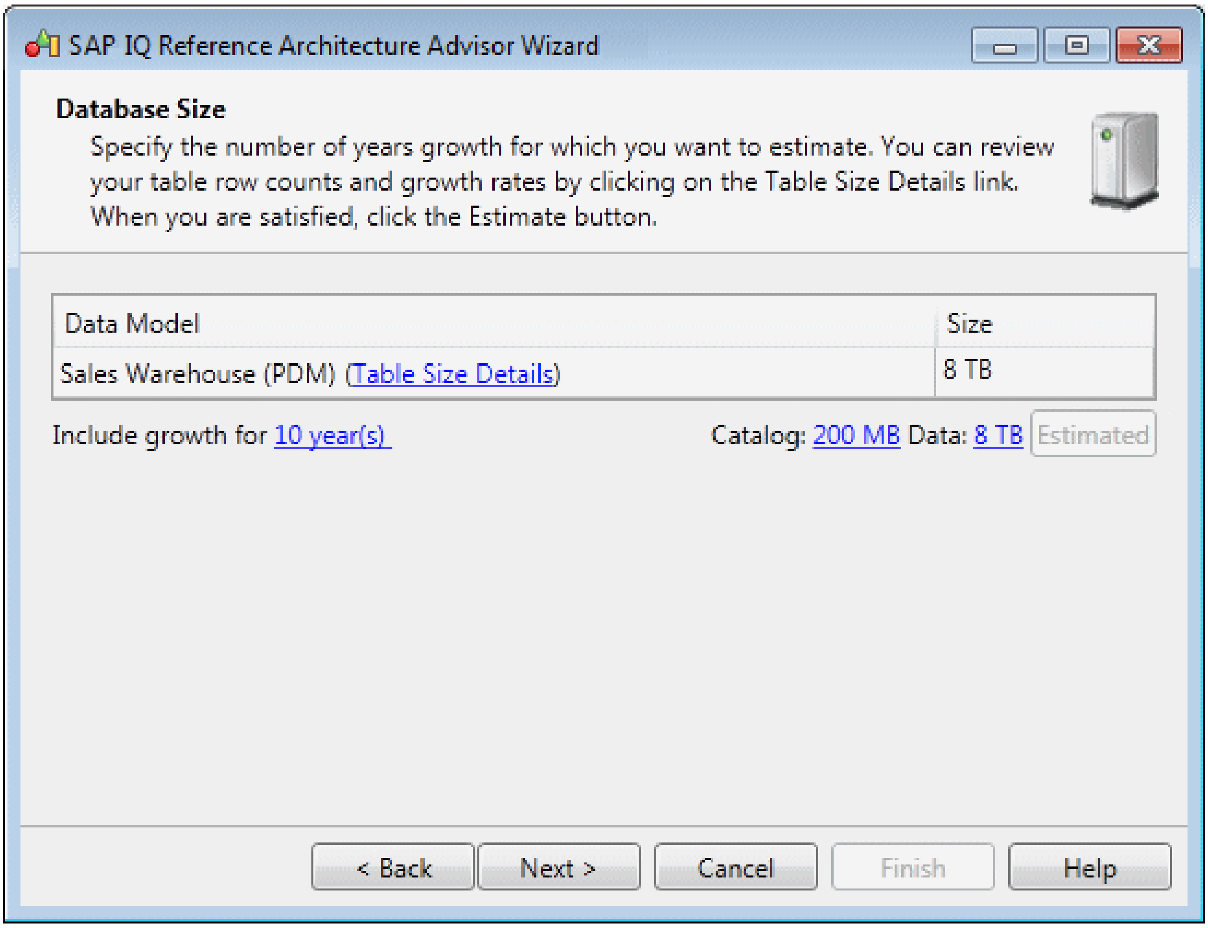
Before launching the advisor wizard, you must develop a physical data model to represent your IQ database and have specified appropriate candidate machines in the hardware profile library (see Hardware Profile Library [page 127]).

Procedure

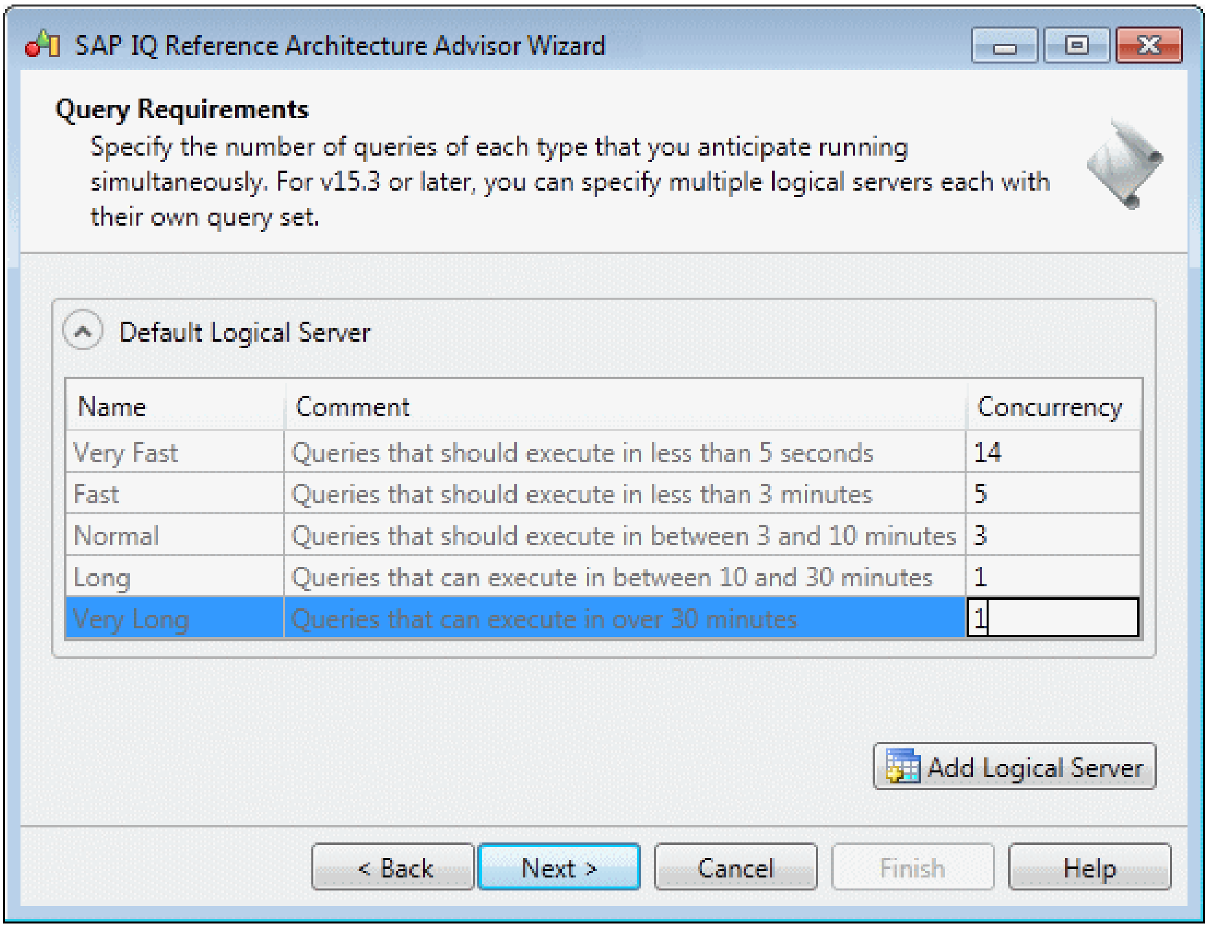
1. Select *Tools* > *SAP IQ Reference Architecture* > *Launch Advisor Wizard* to open the wizard.
2. Click *Next* to go to the *Data Models* page, and use the *Select Data Models* tool to add a PDM from the workspace or the *Select Data Models from Files* tool to open a PDM:



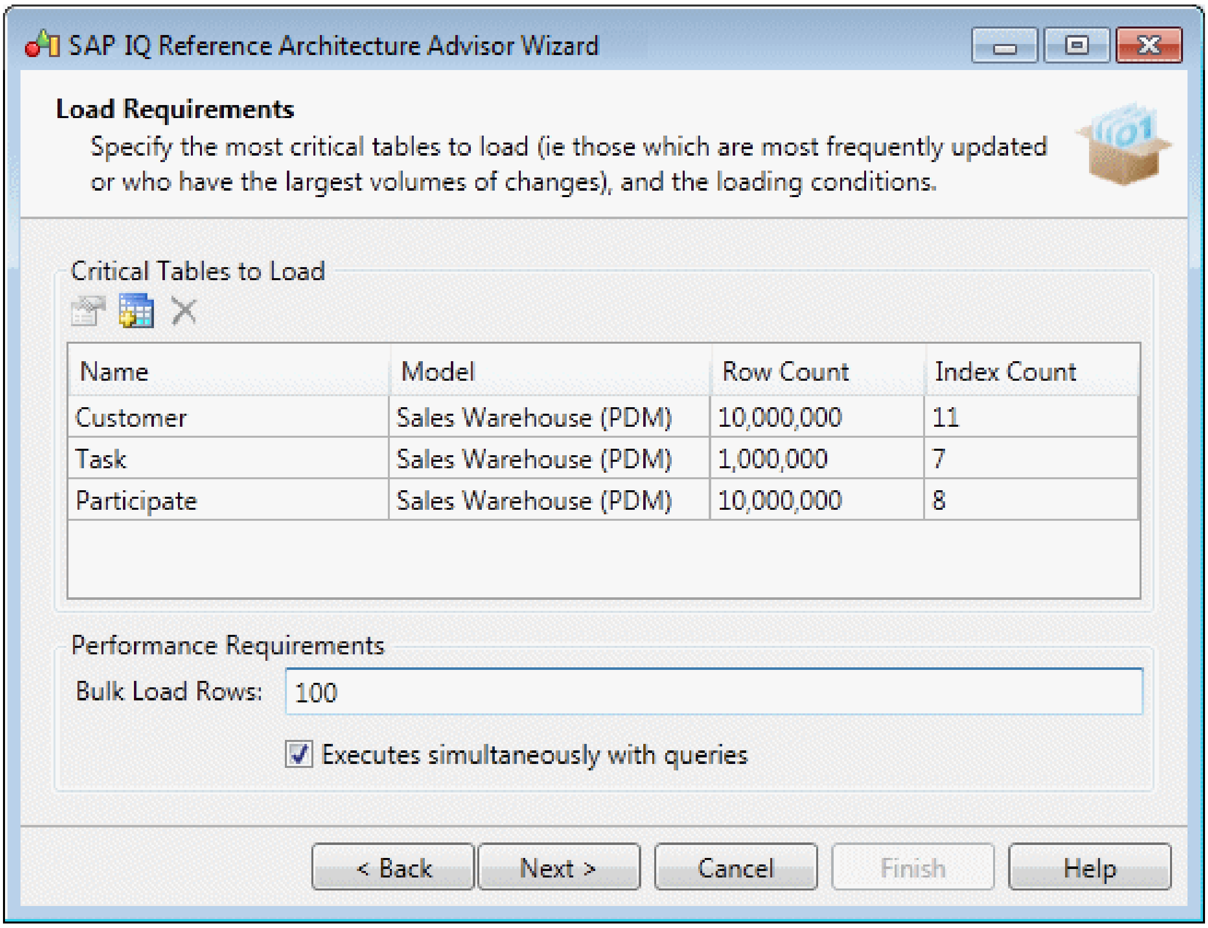
3. Click *Next* to go to the *Database Size* page, and specify the number of years growth for which you want to estimate. To review and edit table row counts and growth rates, click the *Table Size Details* link. When you are satisfied with your figures, click the *Estimate* button to produce an estimate of the overall size of your database at the end of the specified number of years:



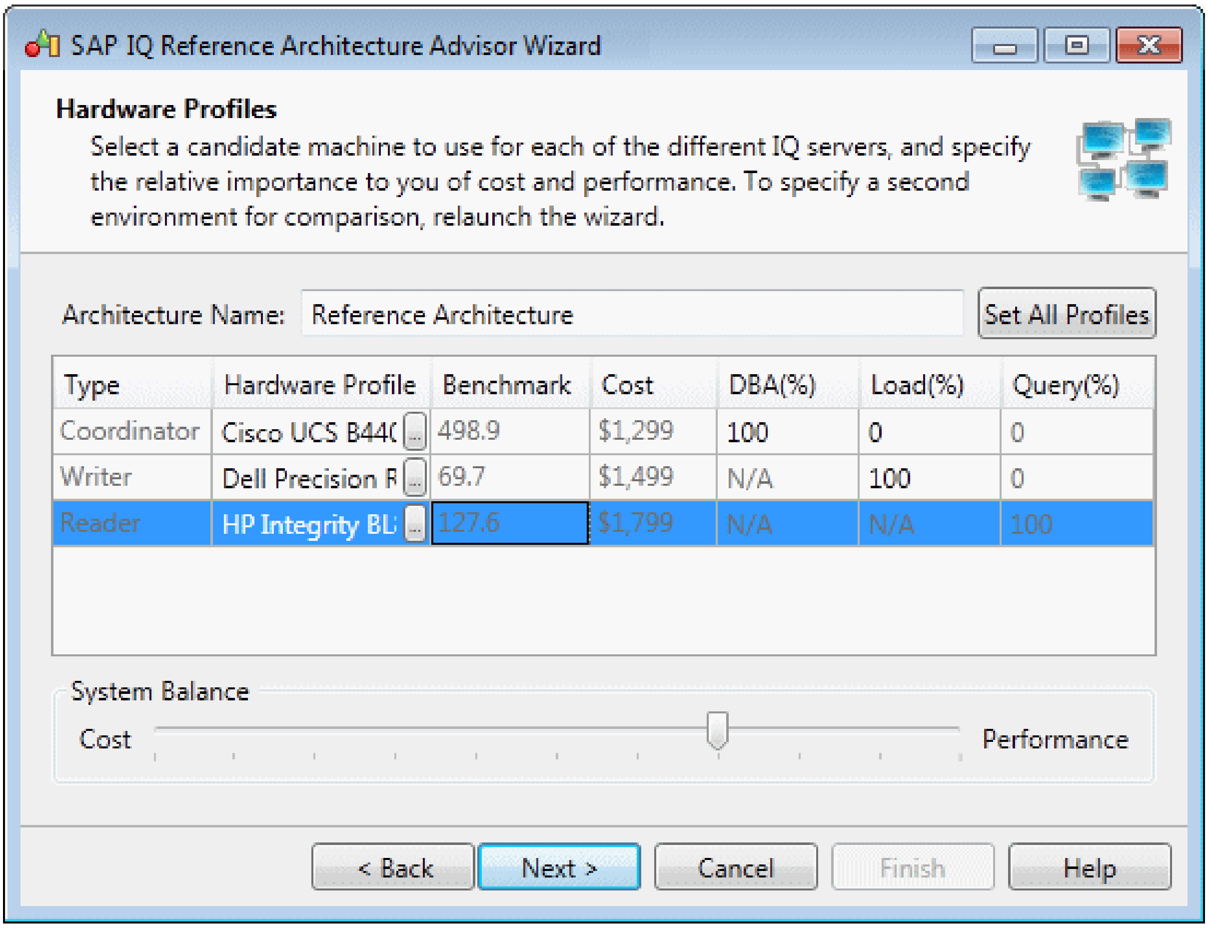
4. Click *Next* to go to the *Query Requirements* page, and specify the number of queries of each type that you anticipate running simultaneously:



5. [optional, v15.3 and higher] Click *Add Logical Server* to define another set of query requirements. You can add as many logical servers as necessary to accurately represent your environment.
6. Click *Next* to go to the *Load Requirements* page. Click the *Select Tables* tool to specify the most critical tables to load (those which have the largest volumes of changes). Then specify the number or rows that will be loaded in batches, and indicate whether this loading will take place while queries are being run:



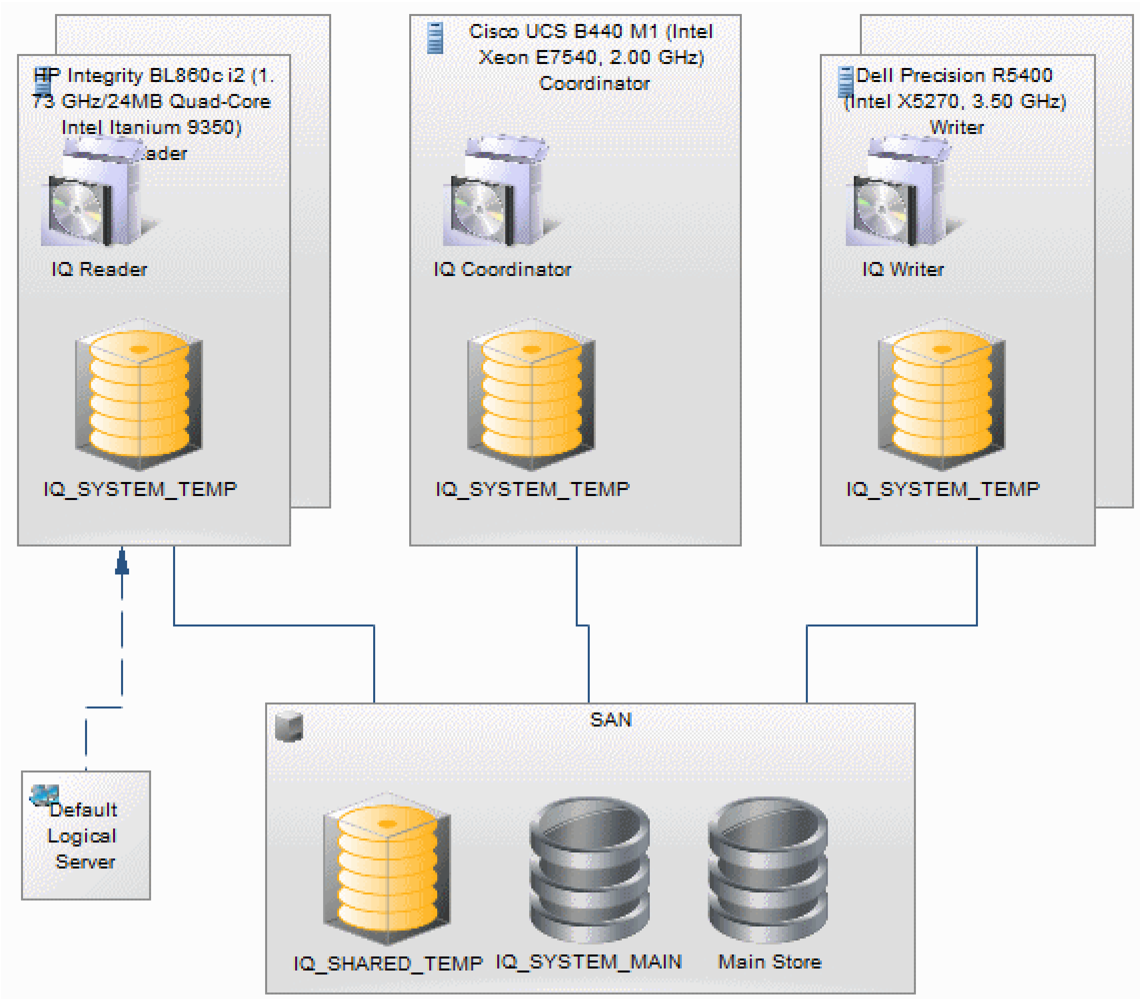
7. Click *Next* to go to the *Hardware Profiles* page. Click the ellipsis tool to select a candidate machine from the Hardware Profile Library for the coordinator, writer, and reader. You can reallocate part of the DBA, load, and query work between these machines as necessary. Use the *System Balance* slider to specify the relative importance to you of cost and performance:



8. Click *Next* to go to the *Summary* page, which lists the information entered on the preceding pages.
9. Click *Finish* to generate your environments.

   The Summary page lists the characteristics of the reference architecture, giving the number of nodes and the total cost of the solution. Click *Back* if you want to change any of your parameters in the wizard.
10. Click *Close* to exit the wizard and go to the generated diagram:

    Double-click the reference architecture symbol to go to a sub-diagram showing the architecture in detail, with separate symbols for the SAN, coordinator, readers and writers:



ℹ Note

To model additional environments for comparison, relaunch the wizard and specify different hardware profiles.

7.2.1 Reference Architecture Properties

To view or edit a reference architecture's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

ℹ Note

If you modify any of these properties (which are generated by the Advisor wizard), you should regenerate the architecture to verify that your environment remains valid (see Modifying and Regenerating Reference Architectures [page 126]).

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Summary	[read-only] Provides an overview of the costs, nodes, and storage sizes of the architecture.

The following tabs are also available:

- *Database Size* – Lists the size of the database, and the period to which the estimate applies. If you change the number of maintenance years, click the *Estimate* button to recalculate the database size.
- *Hardware Profile* – Lists the hardware profiles acting as coordinator, writer and reader, and provides a breakdown of their workloads. The system balance can range between 0% (low cost most important) and 100% (high performance most important).
- *Load Requirements* – Lists the most frequently updated tables and those that have the largest volume of changes, and specifies the IQ loading requirements.
- *Data Models* – Lists the PDMs that represent your IQ database.
- *Query Requirements* – Lists the number of each type of query that must be supported concurrently.

7.2.2 Database Server Properties

To view or edit a database server's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Hardware profile	[read-only] Specifies the hardware profile (see Hardware Profile Properties [page 128]), which defines the server, its cost, the number of instances, and the total cost.

The following tabs are also available:

- *Detail* – [read-only] Provides information about the CPU, cores, memory, storage, file system, and operating system for the server.

- *IQ Servers* – Lists the IQ servers installed on the server (see IQ Server Properties [page 120]).
- *IQ Stores* – Lists the IQ stores defined on the server (see IQ Store Properties [page 121]).

7.2.3 IQ Server Properties

To view or edit an IQ server's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Type	Specifies whether the IQ server is a coordinator, reader, or writer.
Cost	Specifies the cost of the IQ server.

The *Detail* tab contains the following properties:

Property	Description
IQ Page Size	Specifies the page size in KB that is used for the IQ main store and temporary store disk devices.
IQ Catalog Page Size	Specifies the catalog page size in KB for moving data from memory to disk.
Catalog Cache	Specifies the size in MB of memory allocated for catalog operations. The catalog memory will also control, to a degree, how many connections and users can run queries simultaneously in IQ.
Bitmap Memory	Specifies the size in MB of bitmap memory, which is allocated for storing bitmap information.
Main Cache	Specifies the size in MB of the main cache, which is used to hold static, persistent user data and the system structures controlling access to the user data in the form of the IQ indexes.
Temp Cache	Specifies the size in MB of the temporary cache, which is used to hold temporary tables, internal work tables, and any other data structures that are temporary in nature.

7.2.4 SAN Properties

To view or edit a SAN's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Cost	[read-only] Specifies the total cost of the SAN, based on the price specified on the model property sheet (see Reference Architecture Model Properties [page 122]).

The following tabs are also available:

- *Detail* – Lists the sizes of the IQ stores, application store, and backup store, along with the RAID level and access bandwidth in MB/s.
- *IQ Stores* – Lists the IQ stores defined on the SAN (see IQ Store Properties [page 121]).

7.2.5 IQ Store Properties

To view or edit an IQ store's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Store type	Specifies whether the store is a catalog, main, or temp store.
Dbfile size (MB)	Specifies the size of each dbfile in MBs, the number of dbfiles, and the total dbfile size in GBs.

7.2.6 Query Type Properties

To view or edit a query type's properties, double-click its Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Parent object	Specifies the reference architecture for which the query type is defined.
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Concurrency	Specifies the number of this type of query that the environment must be able to perform simultaneously.

7.2.7 Reference Architecture Model Properties

A reference architecture model has more properties than a standard EAM.

The *General* tab contains all the standard model properties (see EAM Properties [page 13]) in addition to the following properties:

Property	Description
Hardware profile library	Specifies the path to the hardware profile library. If no path is specified, the library will be searched for in the `%_LIBRARY%` named path.

Configuration Tab

These properties are available on the *Configuration* tab:

Property	Description
Main cache ratio for reader/ writer (%)	Specifies the percentage of the cache allocated to the main (as opposed to temporary) cache.

Property	Description
Memory margin (%)	Specifies the percentage of memory reserved for the operating system, file system, and other applica-tions.
Application storage (GB)	Specifies the number of gigabytes of storage to be allocated for other applications.
Backup storage ratio (%)	Specifies the percentage of storage allocated to backup data. This number can be greater than 100%.
IQ temp store per user (GB)	Specifies the disk size (in GB) allocated for each concurrent user for the temp store.
SAN disk type	Specifies the type of disk used in the SAN.
SAN disk per core	Specifies the standard and maximum number of disk devices in the SAN system per CPU core.
IQ_SYSTEM_MAIN (GB)	Specifies the standard and maximum sizes (in GB) allocated to the `IQ_SYSTEM_MAIN` dbspace.
Memory per core (GB)	Specifies the standard and maximum memory (in GB) per core.
Main cache per index for writer (IQ page)	Specifies the standard and maximum main cache (in IQ pages) per index for writers.

7.2.8 Logical Server Properties

To view or edit a logical server's properties, double-click its Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Parent object	Specifies the reference architecture for which the query type is defined.
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the ob-ject. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Concurrency	Specifies the number of this type of query that the environment must be able to perform simultane-ously.

The following tabs are also available:

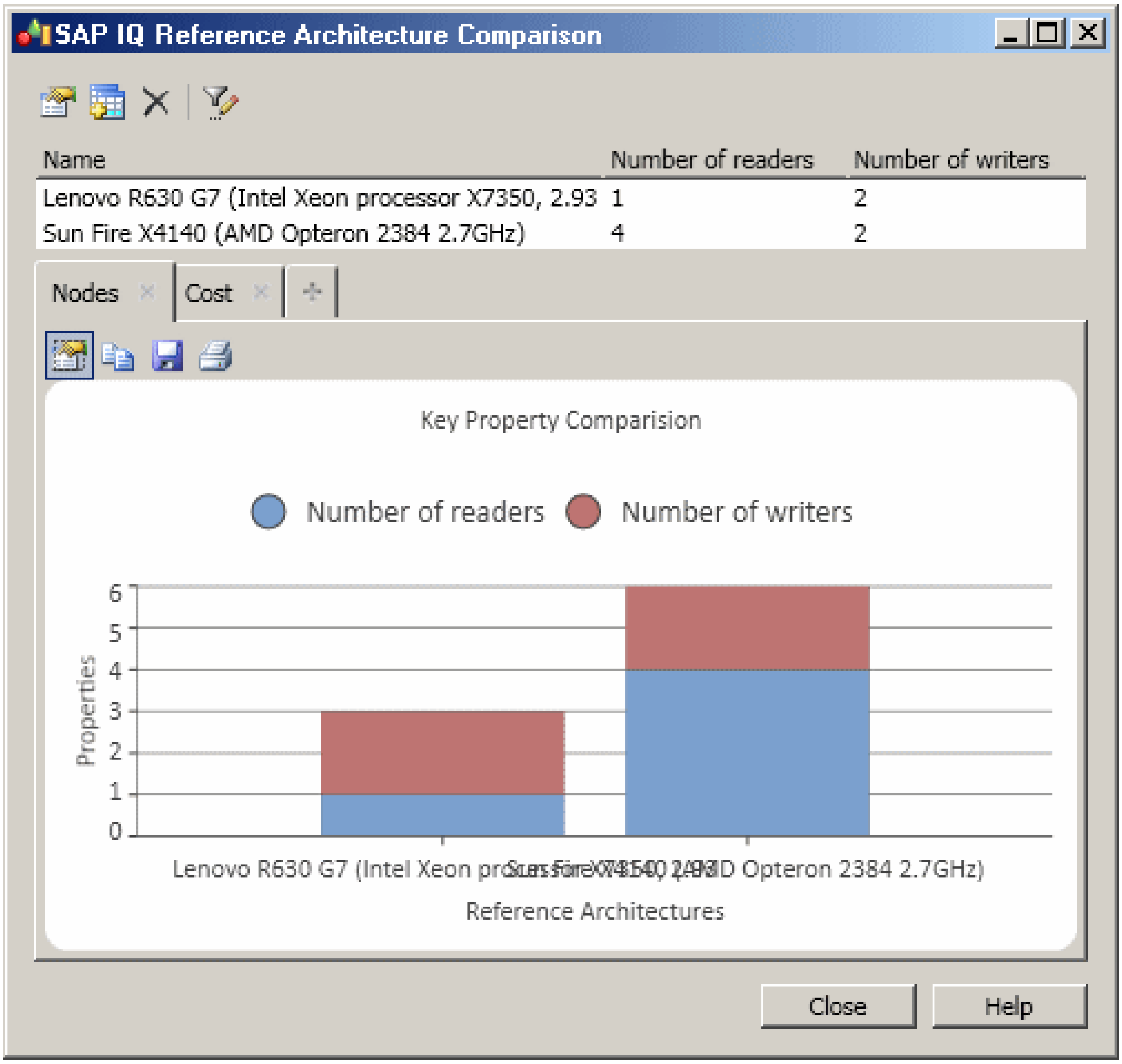
- *Query Requirements* - Lists the number of each type of query that must be supported concurrently.

7.3 Comparing Reference Architectures

PowerDesigner provides a wizard to let you compare any properties of multiple reference architectures using a range of chart styles.

Procedure

1. Select *Tools* > *SAP IQ Reference Architecture* > *Compare Reference Architectures* to open the wizard.
   A default chart using the last chosen parameters appears.
2. [optional] Use the tools above the list to add or remove reference architectures for comparison, and to select properties to display in the list.
3. [optional] Click the *Chart Properties* tool above the chart to select the type of chart and select attributes to compare.
4. [optional] Click the plus sign next to the Chart tab to create addition charts:



5. [optional] Use the other chart tools to copy the chart to the clipboard, save the chart as an image, or print the chart.

7.4 Modifying and Regenerating Reference Architectures

You can modify the properties generated by the Advisor wizard (for example to specify different hardware profiles for reader and writer nodes), but after having done so, you should regenerate the architecture to recalculate the other parameters and verify that your architecture is still valid.

Procedure

1. Modify any of the generated properties on the reference architecture property sheet (see Reference Architecture Properties [page 118]), and then click *OK* to save your changes.

   For example, you could change the number of years for which you want to estimate growth, change the hardware profile for the coordinator, writer, or reader, or modify your query or loading requirements.

   > **Note**
   >
   > If you change the number of years for which you want to estimate growth, you must click the Estimate button to recalculate the database size before regenerating.

2. Right-click the reference architecture in the Browser or diagram and select *SAP IQ Reference Architecture* > *Regenerate*
   A model check is performed to ensure that the architecture is still valid and, if this is the case, it is regenerated.

7.5 Generating Reports from a Reference Architecture Model

You can generate reports from your reference architecture model by selecting *Tools* > *SAP IQ Reference Architecture* > *Generate HTML Report* or *Generate RTF Report*.

8 Hardware Profile Library

PowerDesigner provides a hardware profile library (in the `Library` folder in the installation directory) to store detailed technical information about hardware servers that can be used to model IQ reference architectures or as shortcuts in other models.

8.1 Importing Profiles from the SPEC Website

PowerDesigner provides sample hardware profiles downloaded from the Standard Performance Evaluation Corporation (SPEC) website (www.spec.org), with SPECint2006 Rates benchmark data. You can download additional profiles from the site and import them into your library.

Context

> **Note**
>
> You can also create a hardware profile from the List of Hardware Profiles (*Tools* > *Hardware Profiles*) and enter the properties by hand, but we recommend importing the profiles from the SPEC site.

Procedure

1. In your Web browser, go to the benchmark search form http://www.spec.org/cgi-bin/osgresults?conf=rint2006&op=form
2. Set the following columns to `Display`:
   - Hardware Vendor
   - System
   - # Cores
   - # Chips
   - # Cores Per Chip
   - # Threads Per Core
   - Processor
   - Processor MHz
   - Memory
   - Operating System
   - File System

    ○ Published
3. Enter any appropriate criteria. For example, you may want to restrict the search results to servers from your approved vendors, or to those available since the beginning of the year.
4. Click *Fetch Results* to perform the search.

   The results are returned on a new page.
5. To download all the results, click the *Download* link, next to the Results heading. To download an individual server result, click the *CSV* link to the right of its entry. Both links provide CSV files that can be imported into your hardware profile library.

   > **Note**
   >
   > We advise you to download individual profiles, as they provide more detailed information, including storage size.

6. Right-click the Hardware Profile Library model in the Browser and select *Hardware Profiles* > *Import*.
7. Click the *Select CSV Files* tool and navigate to and select the CSV files you downloaded from the SPEC Website.
8. Click *Import* to import the profiles into the library and then *Close* to close the wizard.
9. Open the property sheets of the imported servers to add the appropriate unit cost to your organization.

   > **Note**
   >
   > If you do not specify costs for your hardware profiles, you will not obtaining useful costing information from the IQ reference architecture Advisor wizard (see Creating Reference Architectures [page 113]).

8.1.1 Hardware Profile Properties

To view or edit a hardware profile's properties, double-click its diagram symbol or Browser or list entry. The property sheet tabs and fields listed here are those available by default, before any customization of the interface by you or an administrator.

The *General* tab contains the following properties:

Property	Description
Name/Code/ Comment	Identify the object. The name should clearly convey the object's purpose to non-technical users, while the code, which is used for generating code or scripts, may be abbreviated, and should not normally include spaces. You can optionally add a comment to provide more detailed information about the object. By default the code is generated from the name by applying the naming conventions specified in the model options. To decouple name-code synchronization, click to release the = button to the right of the *Code* field.
Vendor	Specifies the vendor of the server.
Cost	Specifies the cost of the server. This value is used to calculate the overall cost of the environment.

Property	Description
Benchmark/ Type	Specifies the server's benchmark result and the type of benchmark.

The following tabs are also available:

- *Detail* – Provides information about the CPU, cores, memory, storage, file system, and operating system for the server.

8.2 Hardware Profile Library Properties

A hardware profile library has more properties than a standard EAM.

The *General* tab contains all the standard model properties (see EAM Properties [page 13]) in addition to the following properties:

Property	Description
Currency	Specifies the currency to use in cost calculations.
SAN cost per TB	Specifies the cost of a terabyte of SAN storage.

SAP IQ Configuration Tab

These properties are available on the *SAP IQ Configuration* tab:

Property	Description
Reference Profile	Specifies the hardware profile that is used as a baseline against which to measure the cost and perfo-mance of other profiles.
Load cores per...	Specifies the standard and maximum number of cores required for each column, HG, WD, and other index when loading.
Bandwidth per core (MB/s)	Specifies the standard and maximum disk bandwidth (in MB/s) per core.
Reference Query Baselines	Specifies the supported number of concurrent queries for each query type when the system balance is set to 0% (low cost most important) and 100% (high performance most important).

8.2.1 Specifying Additional Query Types

You can specify additional query types that must be supported by your hardware profiles.

Procedure

1. Open the property sheet of the hardware profile library and click the *SAP IQ Configuration* tab.
2. In the *Query Type Baselines* list, click the *Add a Row* tool, and enter a name and comment to identify and describe your new query type.
3. In the *Minimum Concurrency* column, enter the number of simultaneous instances of the query that the reference profile can support (in addition to all the other query types) when the system balance is set to 0% (low cost most important).
4. In the *Maximum Concurrency* column, enter the number of simultaneous instances of the query that the reference profile can support (in addition to all the other query types) when the system balance is set to 100% (high performance most important).

   The next time that you launch the Advisor wizard in an reference architecture model associated with the hardware library (see Creating Reference Architectures [page 113]), you will be able to specify requirements for the new query type.

9 Checking an EAM

The enterprise architecture model is a very flexible tool, which allows you quickly to develop your model without constraints. You can check the validity of your EAM at any time.

A valid EAM conforms to the following kinds of rules:

- Each object name must be unique in a EAM
- Each network in must be linked to another object
- Each person must belong to an organization unit
- Each software server must be deployed to a physical server

> **Note**
>
> We recommend that you check your enterprise architecture model before generating another model from it . If the check encounters errors, generation will be stopped. The *Check model* option is enabled by default in the Generation dialog box.

You can check your model in any of the following ways:

- Press F4, or
- Select *Tools* > *Check Model* , or
- Right-click the diagram background and select Check Model from the contextual menu

The Check Model Parameters dialog opens, allowing you to specify the kinds of checks to perform, and the objects to apply them to. The following sections document the EAM-specific checks available by default. For information about checks made on generic objects available in all model types and for detailed information about using the Check Model Parameters dialog, see *Core Features Guide > Modeling with PowerDesigner > Objects > Checking Models* .

9.1 Package Checks

PowerDesigner provides default model checks to verify the validity of packages.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.

Check	Description and Correction
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
List of missing terms in object names/codes in package	[if glossary enabled] This special check lists all the terms that are used in the names or codes of all the objects contained in the model or package. • Manual correction: An administrator with write permission on the glossary can double-click the line for this check in the *Result List* to open a merge window in which you can select to add some or all of these missing terms, as appropriate, to the glossary. • Automatic correction: None.

9.2 Application, Application Service, Component and Database Checks

PowerDesigner provides default model checks to verify the validity of applications, application services, components, and databases.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
No deployment instance	The object must be deployed as at least one deployment instance. • Manual correction: Deploy the object as a deployment instance. • Automatic correction: None

9.3 Application Link Checks

PowerDesigner provides default model checks to verify the validity of application links.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code unique-ness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
Unsuitable application link type	The link type is not compatible with the link extremities. For example, if the link type is "use" the link must have a role player as one extremity. • Manual correction: Change the link type. • Automatic correction: Changes the link type.

9.4 Contract Checks

PowerDesigner provides default model checks to verify the validity of contracts.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code unique-ness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.

Check	Description and Correction
Contracts are not correctly used	Contracts must have at least one incoming and one outgoing link with an application service, a business service, or a role. • Manual correction: Create input and output links with the appropriate objects. • Automatic correction: None
Multiple input service types	Contracts must have either roles, business services, or application services as inputs, but not any combination of these types. • Manual correction: Remove any input links that are not of the same type. • Automatic correction: None
Same input and output link	Contracts cannot be the intermediary between the same service. • Manual correction: Modify the input or output link. • Automatic correction: None

9.5 Deployment Instance Checks

PowerDesigner provides default model checks to verify the validity of deployment instances.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
Deployment Instance not deployed	Each deployment instance must be deployed to a platform or server. • Manual correction: Deploy the deployment instance to the appropriate object. • Automatic correction: None

Check	Description and Correction
No deployed object specified	The deployment instance must be an instance of some object. • Manual correction: Specify the object to be deployed in the deployment instance property sheet *General* tab. • Automatic correction: None
Unsuitable Software Server	The deployment instance must be suitable for the server software type to which it is deployed. • Manual correction: Modify the software server type or delete the deployment instance from the software server. • Automatic correction: None

9.6 Form Checks

PowerDesigner provides default model checks to verify the validity of forms.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
No deployment instance	The object must be deployed as a deployment instance. • Manual correction: Deploy the object as a deployment instance. • Automatic correction: None

9.7 Hardware Server, Mobile Device, Network Node and Workstation Checks

PowerDesigner provides default model checks to verify the validity of hardware servers, mobile devices, network nodes and workstations.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
Serial number uniqueness	Serial numbers must be unique in the model. • Manual correction: Modify the duplicate serial numbers. • Automatic correction: Appends a number to the duplicate serial numbers.
Asset number uniqueness	Asset numbers must be unique in the model. • Manual correction: Modify the duplicate asset numbers. • Automatic correction: Appends a number to the duplicate asset numbers.
Mac address uniqueness	Mac addresses must be unique in the model. • Manual correction: Modify the duplicate mac addresses. • Automatic correction: Appends a number to the duplicate mac addresses.
Isolated node	The object must be linked with at least one other node. • Manual correction: Link the object to an appropriate object. • Automatic correction: None

9.8 Network Checks

PowerDesigner provides default model checks to verify the validity of networks.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
Isolated node	The object must be linked with at least one other node. • Manual correction: Link the object to an appropriate object. • Automatic correction: None

9.9 Organization Unit Checks

PowerDesigner provides default model checks to verify the validity of organization units.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.

Check	Description and Correction
Manager not belonging to the managed organization	The manager specified in the Manager field must be listed in the People tab. • Manual correction: Add the manager in the list of people on the People tab. • Automatic correction: Adds the manager to the list of people.

9.10 Person Checks

PowerDesigner provides default model checks to verify the validity of people.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
No organization unit	A person must belong to an organization unit. • Manual correction: Add the person to an organization unit. • Automatic correction: None
Circular dependency through manager property	A person cannot manage herself. • Manual correction: Remove the person from the Manager field of her own property sheet. • Automatic correction: None
Manager must belong to the same organization unit	A person's manager must belong to the same organization unit as her. • Manual correction: Assign the manager to the same organization unit as the person being managed. • Automatic correction: None

9.11 Software Server Checks

PowerDesigner provides default model checks to verify the validity of software servers.

Check	Description and Correction
Name/Code contains terms not in glossary	[if glossary enabled] Names and codes must contain only approved terms drawn from the glossary. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: None.
Name/Code contains synonyms of glossary terms	[if glossary enabled] Names and codes must not contain synonyms of glossary terms. • Manual correction: Modify the name or code to contain only glossary terms. • Automatic correction: Replaces synonyms with their associated glossary terms.
Name/Code uniqueness	Object names must be unique in the namespace. • Manual correction: Modify the duplicate name or code. • Automatic correction: Appends a number to the duplicate name or code.
Software Server not deployed	A software server must be deployed to a physical server. • Manual correction: Deploy the software server in the appropriate object. • Automatic correction: None
Software Server serial number uniqueness	Software Server serial numbers must be unique in the model. • Manual correction: Modify the duplicate serial numbers. • Automatic correction: Appends a number to the duplicate serial numbers.

10 Importing and Linking to and Generating and Exporting from an EAM

PowerDesigner provides various tools to import to and generate and export from your EAM, and to link your EA objects to objects in other models.

10.1 Importing EA Metadata from Excel

While you can manually create objects to represent all the sites, servers, apps and other assets of your enterprise, these kinds of lists or catalogs are commonly collected in spreadsheets and imported to PowerDesigner en masse.

You can import any number of different types of objects from a single Excel file, so long as they can all be imported to a single type of model. Each type of object should be listed in its own table on a separate worksheet of the Excel file. Each row in a table represents one object to import, and each column represents one property (an attribute or list of associated objects) of the object.

A set of nine Excel files containing the data imported to create the objects in the EA Example model are included at <install_dir>/Examples/EAExample. To reuse any of these files for your own imports, simply delete the sample data and enter your own. You can add further columns in a table to import other properties, and add further worksheets to import other types of objects.

The following files are provided:

- `01 Organization Architecture Import.xlsx` - To import sites, organization units, and people (see Importing Organization Architecture Metadata from Excel [page 50]).
- `02 Process Architecture Import.xlsx` - To import process groups, business processes, business capabilities, and areas (see Importing Process Architecture Metadata from Excel [page 64]).
- `03 Application Architecture Import.xlsx` - To import systems, applications, and databases (see Importing Application Architecture Metadata from Excel [page 85].
- `04 Infrastructure Architecture Import.xlsx` - To import hardware servers (see Importing Infrastructure Architecture Metadata from Excel [page 97]).
- `05 Role Associations Import.xlsx` - To import role associations to define people as owners of processes, capabilities, servers, applications, and databases (see Importing Role Associations from Excel [page 46]).
- `06 Deployment Instances Import.xlsx` - To import deployments of applications and databases to hardware servers (see Importing Deployment Instances from Excel [page 95]).
- `07 Goals and Projects Import.xlsx` - To import corporate goals and EA programs and projects (see Importing Corporate Goals and EA Projects from Excel [page 109]).
- `08 Impacts Fulfillments and Stakeholders Import.xlsx` - To import these links between projects and other EA assets (see Importing Impacts, Fulfillments, and Stakeholders from Excel [page 110]).
- `09 Traceability Links Import.xlsx` - To import traceability links between various EA objects (see Importing Traceability Links from Excel [page 148]).

During the import, in addition to the objects that are created, PowerDesigner creates an Excel Import object that contains your import parameters, along with a Table Mapping object for each of the imported tables, which lists the column mappings used. Right-click the Excel Import Object to access the following commands:

- *Import* - to re-import your file based on the options and mappings you have already defined. All the objects will be reimported, overwriting any existing objects based on their names. If you have added new objects to your worksheet tables they will be added to the model, but any objects that you have deleted from your worksheet will not be deleted from the model.
- *Change Mappings* - to re-import your file via the Excel Import Wizard in order to modify your table or column mappings.
- *Change Options* - to change the import options preparatory to re-importing with the command *Import*.
- *Properties* - to open the Excel Import properties sheet.

For detailed information about working with PowerDesigner and Excel, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.

10.2 Importing Visio Diagrams into PowerDesigner

Importing your Visio diagrams into PowerDesigner's rich metadata environment enables you to link your architectural objects with the objects that will implement them, and to profit from PowerDesigner's powerful impact and lineage analysis features. You must have installed Visio 2002 or higher and have selected to install the Visio plug-in from the PowerDesigner installer.

Context

Note

Only Visio diagrams created from the following standard templates can be imported into PowerDesigner, and only objects available on the standard stencils will be imported. Custom properties will be imported as extended attributes.

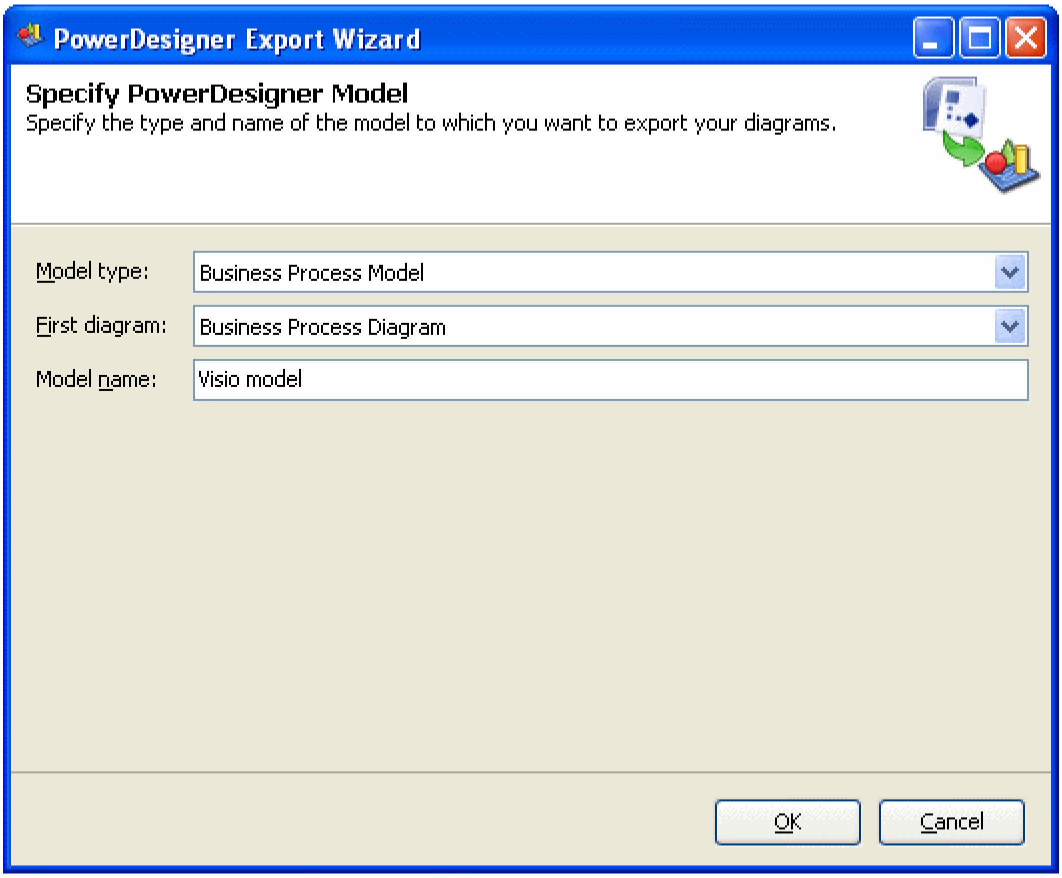
You can import the following diagrams into a PowerDesigner BPM or EAM:

Visio Template	PowerDesigner Diagram
Audit Diagram	BPM Analysis/ Business Process Diagram
Basic Flowchart	BPM Analysis/ Business Process Diagram
Cross-Functional Flowchart	BPM Analysis/ Business Process Diagram
Business Process/ Data Flow Diagram Software/ Data Flow Diagram	BPM Data Flow Diagram

Visio Template	PowerDesigner Diagram
Event Driven Process Chain Diagram	BPM Business Process Diagram
ITIL Diagram	BPM Business Process Diagram
Work Flow Diagram	BPM Business Process Diagram
Flowchart/ SDL Diagram	BPM Business Process Diagram
Organization Chart	EAM Organization Chart Diagram
Software/ Enterprise Application	EAM Application Architecture Diagram
Network/ Basic Network / Detailed Network Diagram	EAM Technology Infrastructure Diagram
Active Directory	EAM Organization Chart Diagram
LDAP Directory	EAM Organization Chart Diagram

Procedure

1. Open your diagram in Visio and select *PowerDesigner* > *Export to PowerDesigner Model* to open the PowerDesigner Export wizard:



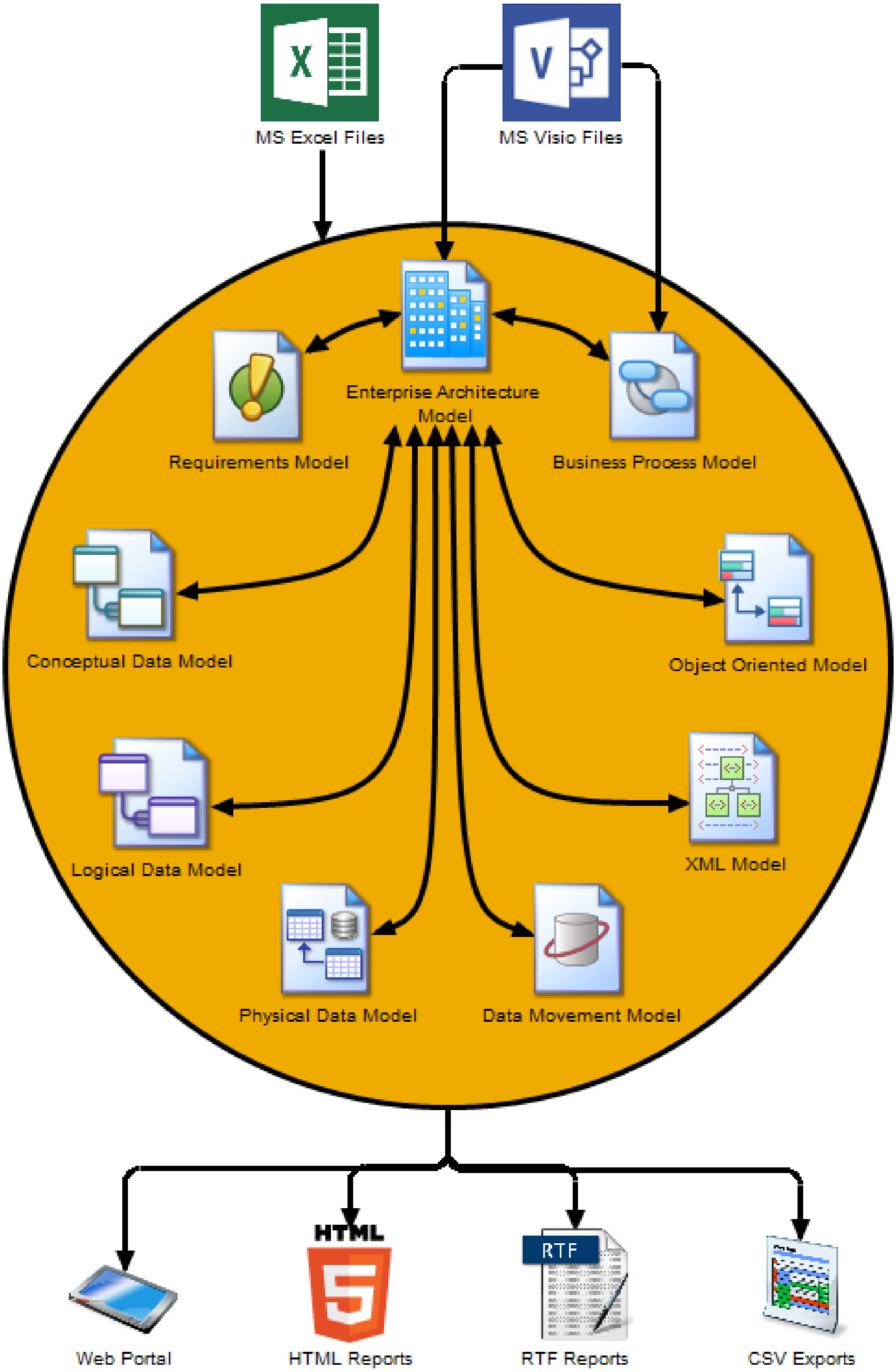
2. Specify the type of model to which you want to export your diagram, enter a name for the model to be created, and then click *OK* to start the export.
3. When the export is complete, click *OK* to close the wizard.

   The diagram is opened as a new BPM or EAM in PowerDesigner.

10.3 Exporting and Importing Objects to and from Other Models

While the EAM provides a big picture of your organization and permits you to decompose your capabilities, processes, and systems to a certain extent, it is not suitable for showing detailed technical implementation information. When modeling the implementation of databases, processes, web services, and other assets, you will use the appropriate specialized PowerDesigner model.

The EAM provides wizards to export EA objects to these other, lower-level models and to import implementation objects into the EAM. Objects exported or imported in this way retain links to the original objects to keep all your models synchronized and to allow you to perform cross-module impact analysis to identify the technical impact of changes on your enterprise architecture:

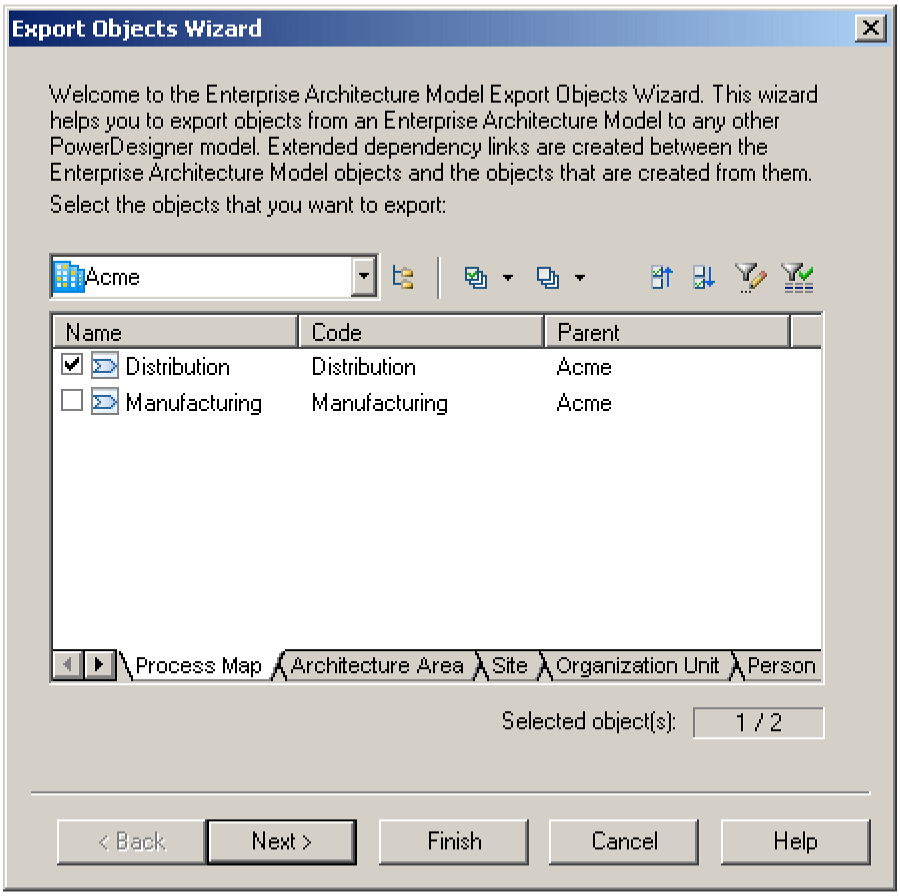

MS Excel Files
MS Visio Files
Enterprise Architecture Model
Requirements Model
Business Process Model
Conceptual Data Model
Object Oriented Model
Logical Data Model
XML Model
Physical Data Model
Data Movement Model
HTML
RTF
Web Portal
HTML Reports
RTF Reports
CSV Exports

10.3.1 Exporting Objects from an EAM to Another Model

The Export Objects Wizard helps you to export EAM objects to other PowerDesigner models to allow you to model their implementation in detail.

Procedure

1. Select *Tools* > *Export Objects* to open the Export Objects Wizard:



2. Specify the EAM objects that you want to export to another model by selecting them from among the sub-tabs in the selection field, and then click Next.
3. Specify the model to which you want to export the EAM objects from among the models open in the workspace, and then click Next.
4. Specify the kind of object that you want to create in the target model.

5. [optional] Enter or select a stereotype to identify the type of dependency that will be created between the EAM objects and the objects that you are creating from them. For example, "Implemented by".
6. Click Finish to start the export.

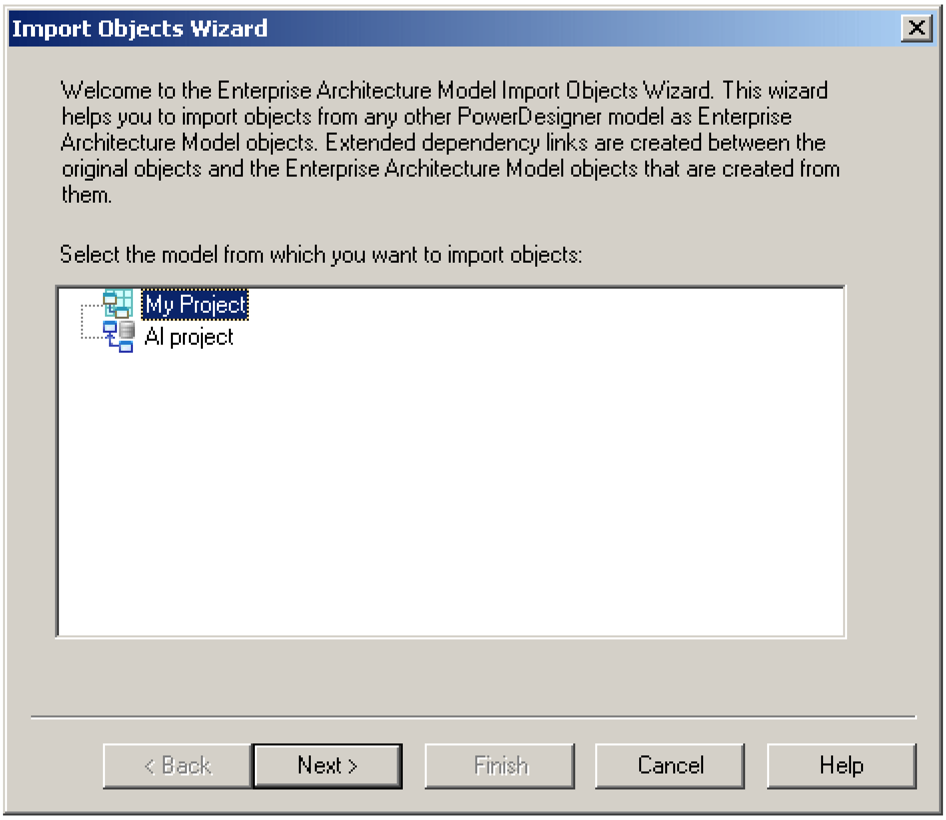
   The EAM objects are exported to and created in the specified model.

10.3.2 Importing Objects from Another Model into an EAM

The Import Objects Wizard helps you to import objects from other PowerDesigner models into an EAM to model their placement in your enterprise architecture.

Procedure

1. Select *Tools* > *Import Objects* to open the Import Objects Wizard:



2. Specify the model from which you want to import the objects to the EAM from among the models open in the workspace, and then click Next.
3. Specify the objects that you want to import to the EAM by selecting them from among the sub-tabs in the selection field, and then click Next.

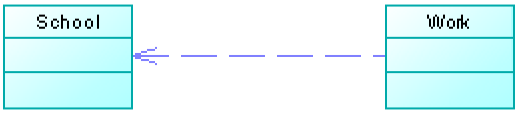
4. Specify the kind of object that you want to create in the EAM and the location (package) where you want to create them.
5. [optional] Enter or select a stereotype to identify the type of dependency that will be created between the objects and the EAM objects that you are creating from them. For example, "Implements".
6. Click Finish to start the import.

   The objects are imported to and created in the EAM.

10.4 Traceability Links

Traceability links provide a flexible means for creating a connection between any object in any type of model and any other object in the same model or any other model open in your workspace. Traceability links have no formal semantic meaning, but can be followed when performing an impact analysis or otherwise navigating through the model structure.

To create a traceability link between objects in the same diagram, select the *Link/Traceability Link* tool in the Toolbox. Click inside the symbol of the object that is dependent and, while continuing to hold down the mouse button, drag the cursor and release it on the symbol of the object on which it depends. In this example, the `Work` entity is shown as being dependent on `School` through a traceability link:



To create a traceability link to any object in any model that is open in the Workspace, open the property sheet of the dependent object, click its *Traceability Links* tab, and click the *Add Objects* tool. Use the *Model* list to select a different model, select the object to point to and click *OK* to create the link and return to the dependent object's *Traceability Links* tab. You can optionally specify a type for any traceability link in the *Link Type* column.

Click the *Types and Grouping* tool to perform various actions on this tab:

- To make a link type available for selection in the *Link Type* column, click the *Types and Grouping* tool and select *New Link Type*. Enter a *Name* for the link type and, optionally, a *Comment* to explain its purpose, and then click *OK*.

  > **Note**
  >
  > Traceability link types created in this way are stored as stereotypes in an extension file embedded in the model. To work directly with this file click the *Types and Grouping* tool and select *Manage Extensions*. For detailed information about working with these files, see *Customizing and Extending PowerDesigner > Extension Files* .

- To control the display and grouping of links, click the *Types and Grouping* tool and select:
  - *No Grouping* - to display all the links in a single list.
  - *Group by Object Type* - to display links to different types of objects on separate sub-tabs. To add a link to a new object type, click the plus sign on the leftmost sub-tab.
  - *Group by Link Type* - to display different link types on separate sub-tabs. To add a new link type, click the plus sign on the leftmost sub-tab.

Note

To see all of the objects that point to an object via traceability links, open its property sheet, click its *Dependencies* tab, and click the *Incoming Traceability Links* sub-tab.

10.4.1 Importing Traceability Links from Excel

You can import traceability links, which can link any objects in your models with a semantic that you define, from lists in Excel files. An Excel file filled with sample data is provided in the PowerDesigner `Examples` folder, which you can adapt for your own imports.

Context

This file forms part of a sequence of files that can be used to create the example model `EA Example`. For information about the other files in the sequence, see Importing EA Metadata from Excel [page 140].

Procedure

1. To create a new EAM to import into, select *File* > *Import* > *Excel File*. Select *Enterprise Architecture Model* from the model type list, enter a **`Model name`**, and then click *OK* to open the Excel Import Wizard.

   Note

   You can import an Excel file into an existing model by right-clicking the model in the Browser and selecting *Import Excel File*. If this option is not available, the Excel Import extension is not attached to your model. To attach it, select *Model* > *Extensions* to open the List of Extensions, click the *Attach an Extension* tool, select **`Excel Import`** on the *Import* tab, and click *OK*.

2. Click the *Select File* tool, browse to and select the Excel file to import, and click *Open* to return to the wizard.

   The example file `09 Traceability Links Import.xlsx` contains three sheets, each of which links objects together via traceability links.

   Note

   Certain metadata in the file cannot be automatically evaluated by the wizard. This file cannot be imported using the *Auto-map columns to properties* option and you must specify the mappings in the wizard.

3. Click the *Options* button set the import options as follows, and click *OK* to return to the wizard:

   - All options - deselected.
   - *Reference associated object by*: **`Code`**
   - *Qualified name separator*: **`.`** (dot)

For detailed information about these options, see *Core Features Guide > Modeling with PowerDesigner > Objects > Importing Objects from Excel Files*.

4. Click *Next* to begin the import.

   Specify how your sheets and columns will be imported in the wizard as follows:
   - **`Trace BC1 to Project`** - Links first-level business capabilities to projects that they own via traceability links:
     1. Import the table as **`Traceability Link`**.
     2. Import the column **`DependentObject`** as **`Dependent Object`** and set the *Object* type to **`Business Capability`**.
     3. Import the column **`LinkType`** as **`Link Type`**.
     4. Import the column **`InfluentObject`** as **`Influent Object`** and set the *Object* type to **`Project`**.
   - **`Trace BC2 to System`** - Links second-level business capabilities to systems on which they depend via traceability links:
     1. Import the table as **`Traceability Link`**.
     2. Import the column **`DependentObject`** as **`Dependent Object`** and set the *Object* type to **`Business Capability`**.
     3. Import the column **`LinkType`** as **`Link Type`**.
     4. Import the column **`InfluentObject`** as **`Influent Object`** and set the *Object* type to **`System`**.
   - **`Trace P4 to System`** - Links fourth-level processes to systems on which they depend via traceability links:
     1. Import the table as **`Traceability Link`**.
     2. Import the column **`DependentObject`** as **`Dependent Object`** and set the *Object* type to **`Business Process`**.
     3. Import the column **`LinkType`** as **`Link Type`**.
     4. Import the column **`InfluentObject`** as **`Influent Object`** and set the *Object* type to **`System`**.

   Progress is displayed in the *Output* window. When the import is complete, a dialog will appear showing how many objects have been created. Click *OK* to return to your model and review the imported objects.

10.5 Specifying Diagrams as Related Diagrams

You can use related diagrams to further define the behavior and implementation of objects and to view them from different angles and in terms of semantic relationships. You can associate any type of diagram open in the workspace with an object, including diagrams from other packages or models.

To specify that a diagram is a related diagram of an object, open the property sheet of the object and click the *Related Diagrams* tab. If this tab is not visible, click the *More* button. The following tools are available:

Tool	Description
	*Properties* - Opens the property sheet of the selected diagram.

Tool	Description
	Open Diagram - Opens the selected diagram. **Note** If the object is not composite (and so does not have any child diagrams), you can navigate to the first related diagram in the list from an object symbol by pressing *Ctrl* and double-clicking the symbol. To navigate to any related diagram in the list, including for composite objects, right-click the symbol and select *Related Diagram* > `<Diagram Name>`.
	Add Objects - Opens an object selection dialog to select existing diagrams and add them to the list.
	Create an Object - Creates a new diagram, and adds to the list as a related diagram. You must specify the location (model) and type of diagram to create and give it an appropriate name. **Note** To create a new diagram and associate it to the object from the object symbol, right-click the symbol and select *Related Diagram* > `<New>`.

10.6 Generating an EAM from an EAM

You can generate another EAM from your EAM. When changes are made to the source model, they can be easily propagated to any generated models by regenerating, using the Update Existing Model generation mode.

Procedure

1. Select *Tools* > *Generate Enterprise Architecture Model* to open the EAM Generation Options Window:
2. On the *General* tab, select to generate a new or update an existing model, and complete the appropriate options.
3. [optional] Click the *Detail* tab and set any appropriate options. We recommend that you select the *Check model* checkbox to check the model for errors and warnings before generation.
4. [optional] Click the *Target Models* tab and specify the target models for any generated shortcuts.
5. [optional] Click the *Selection* tab and select or deselect objects to generate.
6. Click *OK* to begin generation.

Results

> **Note**
>
> For detailed information about the options available on the various tabs of the Generation window, see *Core Features Guide > Linking and Synchronizing Models > Generating Models and Model Objects*.

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: https://help.sap.com/viewer/disclaimer).

go.sap.com/registration/contact.html