

BACONMAP.ORG

# Baconmap Technical Document

---

Resources Defined

**Baconmap Crew**

**5/9/2009**

## Table of Contents

BaconMap .....	3
BaconMap Philosophy .....	4
Features and Benefits .....	4
General Dependency Rules .....	5
Security .....	5
Anatomy of a Baconmap.....	6
Resource Panel .....	7
Box .....	8
Server .....	8
Service.....	8
Database .....	9
Application.....	9
Device .....	10
Group .....	10
Resource Tree.....	11
Menu .....	12
Work Panel.....	17
Appendix .....	21

## BaconMap

BaconMap is IT resource dependency mapping software. Its goal is to assist IT administrators and managers with the complexity of today's IT systems. The advent of virtual servers and virtual websites has added a level a complexity to IT resource management by hiding resources within resources. This software helps IT personnel define their existing network and system, then generate dependency maps based on the information.

BaconMap has multiple goals: dependency mapping, resource tracking, information repository, and IT strategy and disaster recovery planning. The first goal is to provide IT personnel a visual representation of their networked systems and to simplify the tracking of complex dependencies. A second goal is to provide an information repository to be used by the IT department, much like a wiki. The final goal is synthesize the information and deliver it in manner facilitating short and long term IT planning, to include disaster recovery planning.

BaconMap's dependency map represents a cause and effect map. The objective is to identify relationships between IT resources based on type of dependency they have and not just a network relationship. BaconMap attempts to answer the questions: what happens when a particular resource fails? What else fails? What are the critical systems I should be strengthening based on the dependencies?

Key to incident management is proper and expedient communication with the resource stakeholders; people who depend on the resource. Preparing an action plan pre or post incident requires the input from the users of the resource and a clear understanding of what happened and how to resolve the problem. BaconMap allows IT administrators to easily track points of contact for every resource. This gives IT personal a scope of the impact as it pertains to people; which is important in some highly politicized environments. Additionally, each resource can act as its' own data container; a repository for key information; such as information critical to maintaining the resource, vendor information, special instructions, and images, to name a few.

BaconMap, in the role of a wiki, can attach resource specific documents to any object defined. Documents like vendor invoices, stopping and starting procedures, helpdesk notes, etc can be attached to any resource. This allows IT related information to be securely centralized and available to IT members at all times. This is especially helpful during times when IT administrators are gone for a period of time and an issue arises that need immediate resolution. Other information like purchasing cost and annual support cost can be assigned to each resource independently. The BaconMap reports can quickly show when support and maintenance are due and their costs. A summary cost inventory can also be created for any resource type, which is helpful during the budgeting process.

BaconMap supports response time objectives (RTO) aiding in the disaster recovery planning. A resource defaults to a 30 day RTO but can be set as low as 0 days. The RTO report will generate a listing of resources by lowest RTO to highest. This allows IT management to list critical resources and design sensible disaster recovery plans within context of the entire IT infrastructure.

At its very foundation BaconMap is a teaching tool. Once populated, the dependency maps, reports and uploaded documents make BaconMap an instructional website. New hires can quickly learn the complexities of the network resources, managers (low, middle, upper) can view summarized reports and maps of the network assisting them in the decision making process. Perhaps most important, critical resources are quickly identified and pro-active decisions can be made, potentially saving the organization time and money, and increasing overall uptime.

## BaconMap Philosophy

Bacon map was written with a genealogical philosophy in mind with a communal tone. Children (resources) can have many parents and parents (resources) can have many children. The failure of a parent results in the failure of its child. As an example, a child having multiple parents identifies critical points of failure that could be a potential problem in the future.

BaconMap is not a network and system monitoring tool and such does not automatically populate itself. BaconMap is an information repository and requires input and maintenance from all levels of IT. It is a 'community tool' that collects important information from the combined IT minds in the department. The resource level design facilitates high-level granularity mapping accepting input from all facets of IT support. As community tool all members of the IT department add and edit information thus generating a smarter, more knowledgeable team, and more efficient IT department.

The focus when building a dependency graph should be on creating critical dependencies between resources. This means focusing on the failure related dependencies more so than defining an exhaustive list of dependencies. For example, a web server has an electrical dependency from the building but more importantly it has a router/network dependency which may be more apt to fail than the electricity. Also a building electrical failure impacts many more resources than a network/router failure and constitutes a far more critical problem for all resources, where as a network/router failure is only critical to a few key resources. The objective is to analyze your network and identify those critical paths that are more than likely to be problems in the future and map them.

### Rules & Resources

There are 7 types of resources BaconMap tracks: box, server, service, database, application, groups and device. These resources are specific to network and server systems found in a typical IT shop. Each resource has its own rules by which they can be interconnected, edited and removed. The rules allow BaconMap to correctly map relationships and identify critical points in the map.

## Features and Benefits

BaconMap is designed to work like a wiki with relationship maps. Resources are linked to each other based on dependencies, which helps IT administrators and managers visually assess the state of their networked systems based on **their** definition of critical paths and dependencies. This makes their maps very unique to each site. In addition to the map each resource is a repository of its' own system information. This allows IT members to quickly find critical information for each resource. The mapping and wiki components provide a wealth of options to the IT team.

**Training:** By mapping critical resources and their dependences, the IT staff can quickly learn what the critical systems are for their environment. This is beneficial for new hire training and for helpdesk personnel trouble shooting failures; whether they are email related, dhcp, website, etc. The repository for each resource can provide helpdesk staff critical information or procedures regarding stopping and starting an application or server.

**Planning:** The *big picture* of a well maintained map quickly identifies critical resources and potential problems. Resources with heavy dependencies are reflected in darker red than those with few dependencies. IT managers can use this information to plan resource redistribution, load balancing or justification for additional resources.

**Centralization:** The Wiki component allows managers to attach important documents to the resource, such as vendor information, purchasing information and system configuration documents. Key personnel can now offload important information in their brain unto BaconMap and share it with the IT team, lessening the problems with missing personnel. The document encryption feature can protect accounts and passwords so systems don't have to be broken into or re-imaged when a password is forgotten.

**Management:** BaconMap can inform the IT staff of the impact of the failure of any given resource. This provides management information as to who to contact, the severity of the impact and resources affected. Reports can be generated highlighting the scope of the impact, whether local or impacting the greater community.

**Reports:** Reports can be produced for management showing resources and their dependencies. A list of POCs can be maintained; important for an IT shop managing resources for multiple departments. Other reports can track resources by role or acquisition date.

## General Dependency Rules

These rules apply to all the resources defined in BaconMap. Some resources will have additional rules governing their associations with other resources and are defined in their respective sections.

Adding resources is done top down; a parent needs to exist before a child can be added.

Resources cannot have double lineage dependencies. This refers to resources having each other as parents and children. Dependencies flow in one direction only, so a child a few levels removed from the top level cannot have a child that is already defined as a parent.

BaconMap does not support orphans. Deleted nodes with children will be orphaned and removed if the child does not have other parents linking to it. BaconMap will alert the user if a child will be orphaned prior to executing a resource reassignment or deletion.

## Security

Users log in using their email address. The email address is used when resetting user account passwords.

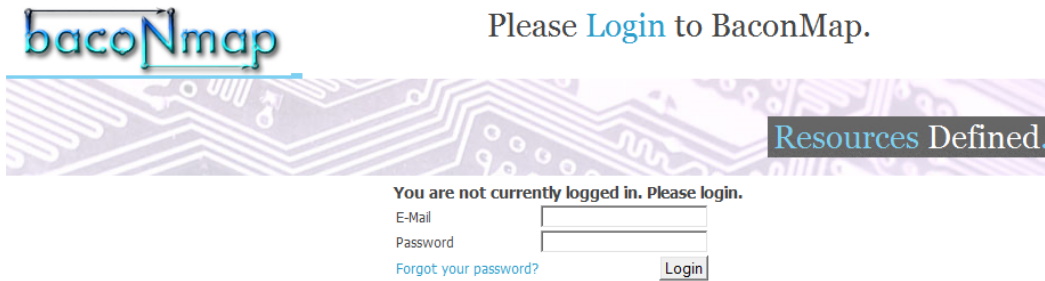


Figure 1 Log In Screen

The logout link is located at the very bottom of any BaconMap page.



Figure 2 Log out

BaconMap supports a 3 access level security mechanism: administrator, read/write , read-only. Access levels are managed via the **admin** tab when logged in as administrator.

A Read-Only account can list the resources and generate the map but cannot change anything. Resource attached documents are viewable to include encrypted documents, if the user knows the encryption password.

Read/Write accounts can add and modify resources to BaconMap, to include managing resource attached documents. System setting changes are restricted to the administrator account.

The administrator account can do everything on BaconMap: account management, resource management, database management.

## Anatomy of a Baconmap

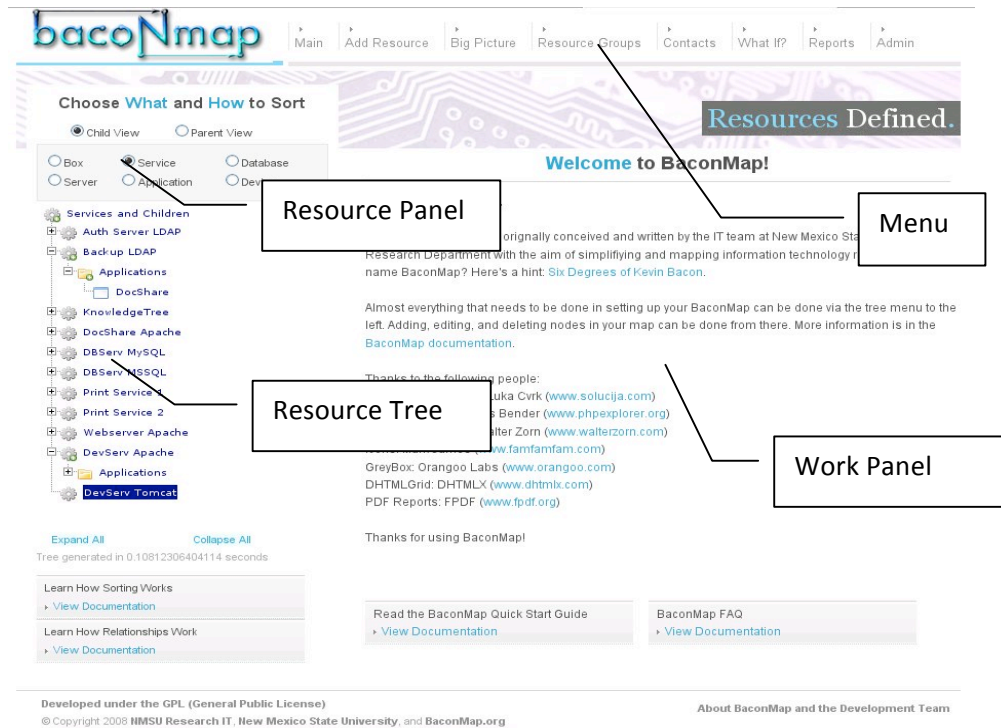


Figure 3 Baconmap

Baconmap was written with Php, Mysql, GraphViz, and EzComponents, all open source software. The application consists of 4 main work areas. The bulk of user interaction will occur on the work panel and the resource tree area.

## Resource Panel

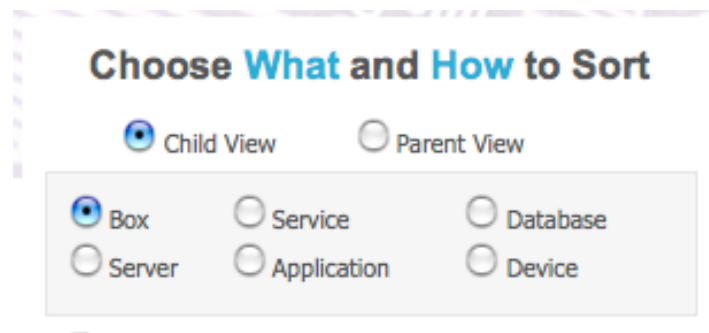


Figure 4 Resource Panel

The resource perspective view controls the resource tree list from a particular user perspective. The child view lists children below the resource type selected. If the *service* type is selected then the tree listing will consist of all services defined in Baconmap and the children attached to them. Conversely, if *Parent View* is selected the listing will consist of the same services but the dependencies will list parents in the tree structure.

The resource type view will place the resource type at the top of the tree listing with corresponding children attached. There are 6 resource types defined: box, service, database, server, application and device.

## Box

A box is hardware that runs an operating system and will host either, services, databases and/or applications. The box can be defined by its network name, manufacturer and model number, or some alias meaningful to the IT shop.

### Parent Rules

1. A box has the option of depending on nothing.
  - a. ONLY device(s) can be parents of a box but are optional
  - b. The box is defined as self-contained and only susceptible to internal failure.
2. A box can NOT depend on another box

### Child Rules

1. A box can only have a server as a child
2. Theoretically a box could have multiple children (ie servers)

## Server

A server is an operating system running on a previously defined box. The operating system types can be defined prior to assigning the OS to a box via the administration tab. The OS can be a typical OS like Linux or Windows 2003, but can also be an OS supporting virtual servers like VMware's ESX OS.

### Parent Rules

1. A server must have a parent of either of these:
  - a. A box can be a parent
  - b. A Virtual server can be a parent
  - c. A device can be a parent (Typically they're not the single parent)

### Child Rules

1. A Server can have the following children
  - a. Server
  - b. Service
  - c. Database
  - d. Application

## Service

A service provides a service/function to a child. The service could be an application like Microsoft Exchange whose only task is to manage email and receive and send data for others. A service is different from an application in that a service does not have a direct end user. Most services assume the duties of the middleware whose functionality is mostly behind the scenes.



## Parent Rules

1. A server can be a parent
2. Another service can be parent
3. A device can be a parent (Typically they're not the single parent)
4. A group can be a parent

## Child Rules

1. A service can have the following children:
  - a. A service can be a child
  - b. A database can be child
  - c. An application can be a child
  - d. A device can be a child

## Database

A database is straight forward. The database is the set of data available to a child. This should not be misconstrued with the software; MySQL and MS SQL are services and not databases. Multiple databases will be listed individually.

## Parent Rules

1. A server can be parent
  - a. Rare
2. A service can be parent
3. A device can be parent
4. A group can be a parent

## Child Rules

1. A service can be a child
2. An application can be a child
3. A device can be a child
  - a. Rare

## Application

An application is the end node in the dependency graph. An application utilizes resources provided to it by all its parents, grand-parents, etc. The application does need a person as an end user it simply justifies the existence of all the parents linked to it, key to this philosophy is that applications cannot have children.

## Parent Rules

1. A service can be a parent
2. A database can be a parent
3. A device can be a parent

## Child Rules

1. No children allowed

## Device

A device is an independent resource which facilitates the operation of a child. Typical devices are SANs, firewalls, RAID box, tape library and UPS. Devices can be located at any level of the dependency map. A device can be defined as a unit made up of set of interdependent devices. A device can also be broken down into its individual key components where one component is the parent of another component, defined in serially. For example a tape library could be defined as the tape drive unit as the parent of the tape library enclosure. A failure of the drive unit signals a failure of the enclosure as well. Conversely, the failure of the tape library enclosure does not mean the drive unit has failed as well.

### Parent Rules

1. A device can have these as parents
  - a. A box can be a parent
  - b. A server can be a parent
  - c. A service can be a parent
  - d. A device can be a parent

### Child Rules

1. A device can have these as children
  - a. A box can be a child
  - b. A server can be a child
  - c. A service can be a child
  - d. An application can be a child
  - e. A database can be child

## Group

A group is not a specific resource but rather a grouping of redundant resources. Redundant resources are defined as those resources providing redundant services, such as multiple active directory servers or clustered Exchange servers. The objective is to define these groups and attach them to children such that a failure of a resource within the group does not render the child ineffective. BaconMap will process the links and as long as there are others in the group still active the child is listed as operational.

### Parent Rules

1. A group can NOT have parents, as it is made up of previously defined resources

### Child Rules

1. A group can have these as children
  - a. A service can be a child
  - b. A database can be child
  - c. An application can be a child
  - d. A device can be a child

## Resource Tree

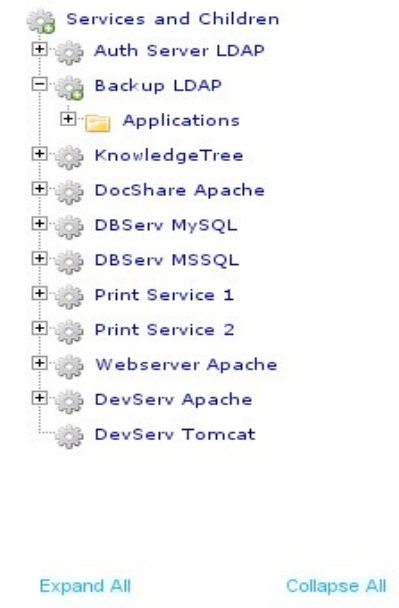


Figure 5 Resource Tree

The resource tree panel will list resources based on the sort and type selected in the resource panel. The resource lists dependencies attached to it, either as a parent view or a child view. A parent view lists parent(s) it depends on, and a child view will list children that depend on it. The design of the application can list the same child or parent defined multiple times under different resources. The resource focus (box, service, application, etc) will list the unique resources at the root level but all dependencies below can have duplicates.

Clicking on a resource will pop-up a small window with basic key information such as description and role in the organization, and most importantly the point of contact for the resource.

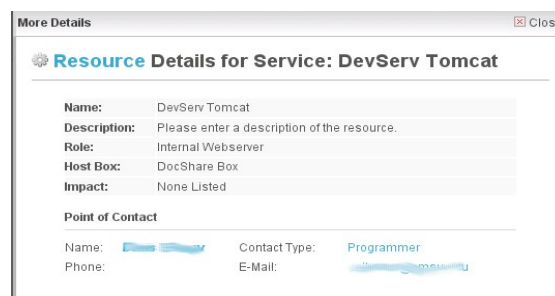


Figure 6 Resource Pop-Up

Clicking on the icon attached to the resource will pop-up a smaller windowing documents additional options.

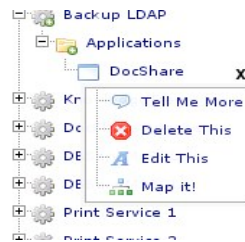


Figure 7 Utility Screen

**Tell Me More** – This is the same screen described in the previous paragraph.

**Delete This** – Deletes the resource and all dependencies to children. Orphaned children will be deleted.

**Edit This** – The will allow changes to the resource information and dependencies. The edit screen will open to the right of the tree list.

**Map It!** – This will generate a map of all the dependencies attached to this specific resource. The map will display in the work area to the right of the tree list.

**Documents** – When documents are attached to the resource, the 'Documents' menu option will be available. When selected a new window will open listing all the available documents for the resource.

## Menu

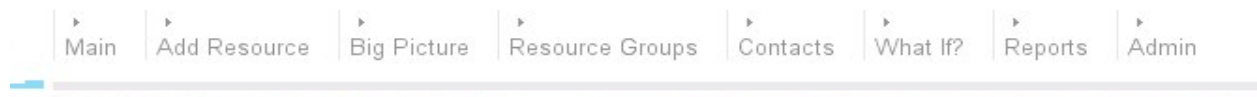


Figure 8 BaconMap menu

**Main** – Displays the default page.

**Add Resource** - Allows the user to add a new resource and assumes no parentage; adding a resource from the resource tree assumes a parent relative to the click.

**Big Picture** – This generates a dependency map of all the resources defined. The map will open in a new window or tab; depending on the browser.

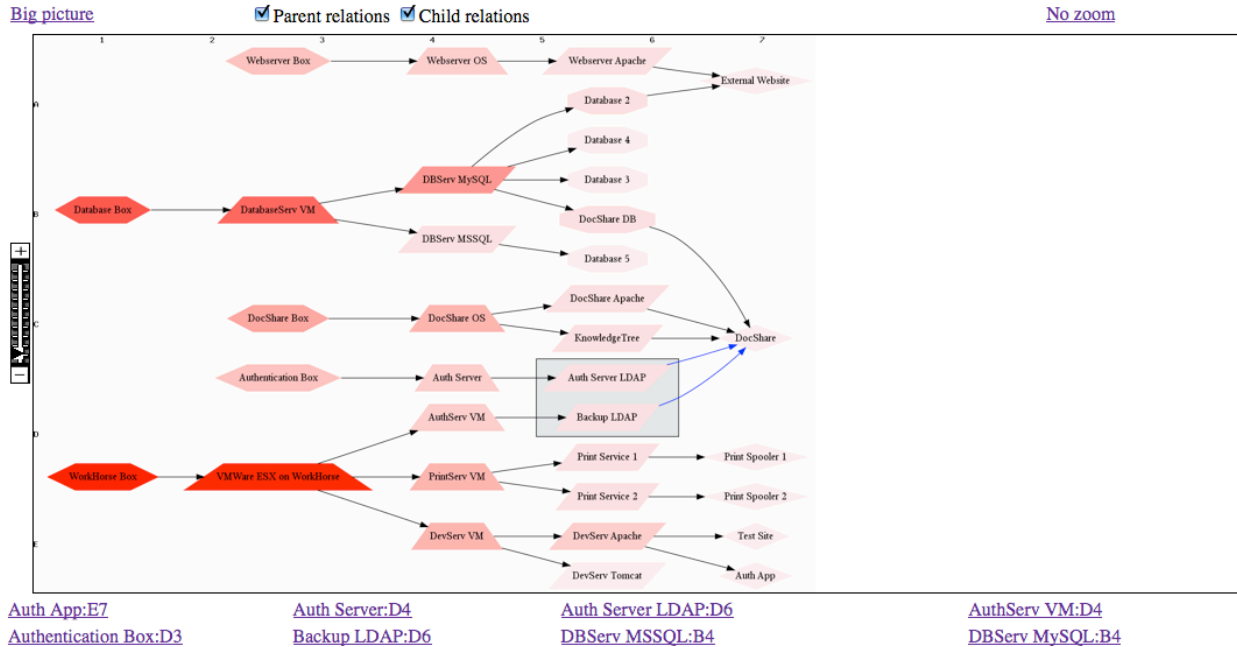


Figure 9 The Big Picture

The map can be zoomed in or out with the mouse wheel. The grey blocked areas denote services that are grouped together and provide redundant support to another resource.

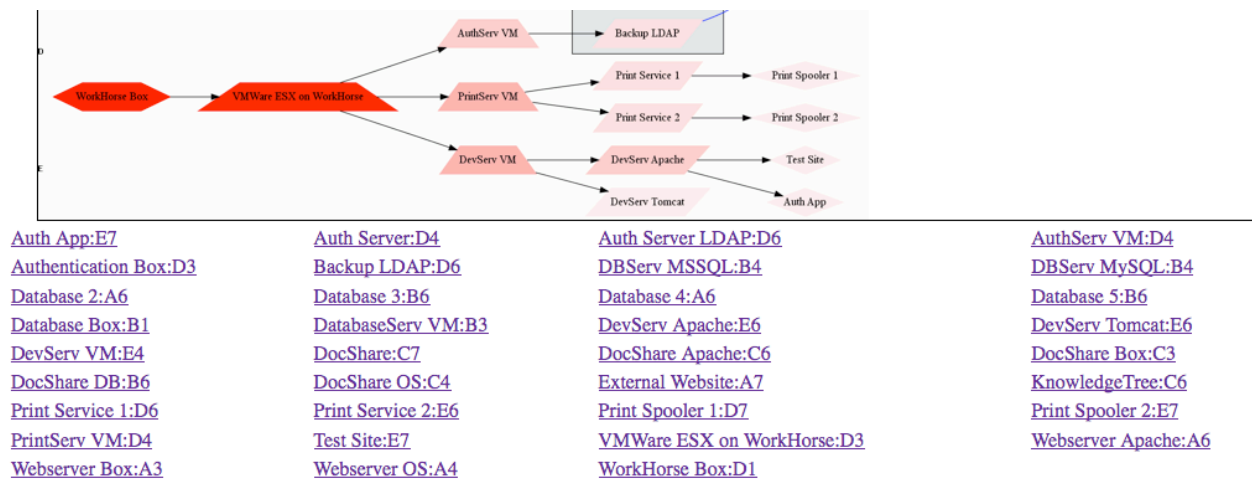


Figure 10 Map Index

Below the 'big picture' is an index of the map. The index is useful for finding a resource in a very busy map or when the 'big picture' is printed.

Holding down the left mouse button will allow you to move the map in any direction. This is handy when the map gets unwieldy.

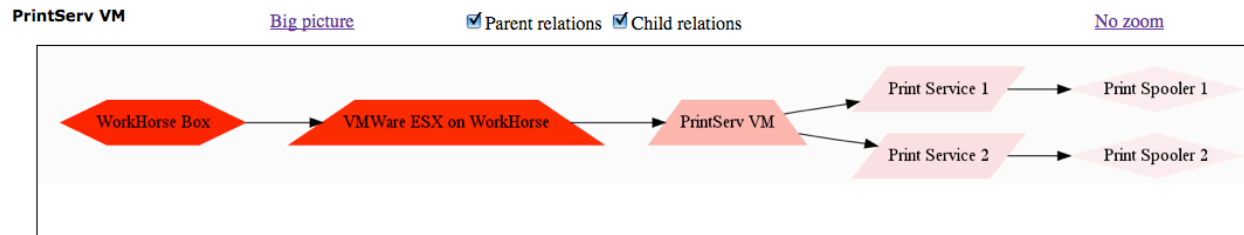


Figure 11 Pruned Map

An added feature is the self pruning affect: clicking on any shape will redraw the map with that resource as the focal point showing only its parents and children. In this example the “PrintServ VM” resource was selected.

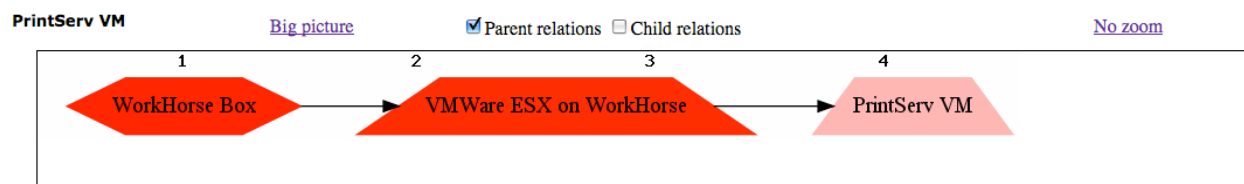


Figure 12 Parents Only

Deselecting the ‘Child relations’ will prune the map to a parent view only from a “PrintServ VM” viewpoint.

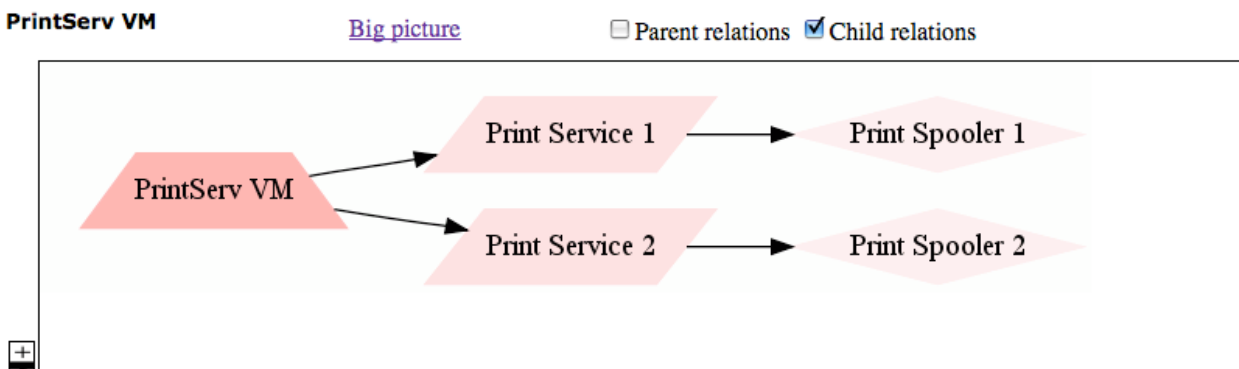


Figure 13 Children only

Deselecting the ‘Parent relations’ will prune the map to a children view only from a “PrintServ VM” viewpoint.

Clicking on the *Big Picture* will regenerate the entire image.

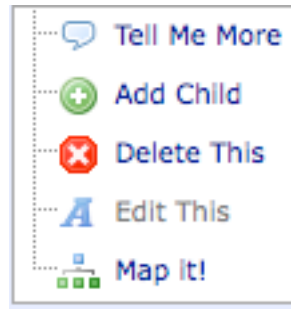


Figure 14 Resource Tree Menu

Clicking on the icon in the resource tree can also generate a resource centric map by selecting “Map It”. This will open a new window with the map.

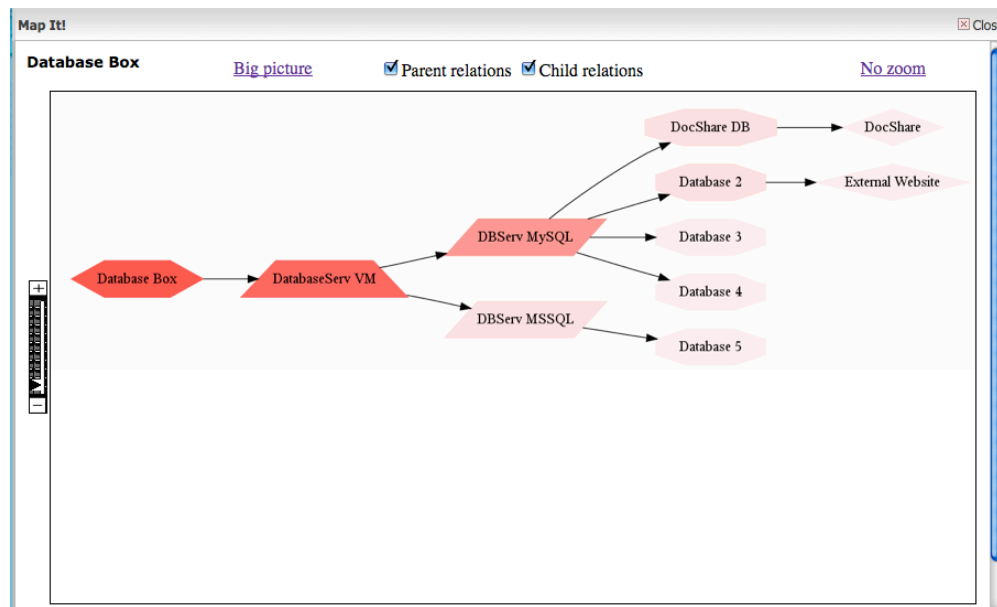


Figure 15 The Little Picture

**Resource Groups** – Allows the user to define redundant/clustered resources. This is useful when a failure of a specific resource is covered by the continued operation of another resource. The map will highlight these types of dependencies in a different colored line.

**Contacts** – Existing points of contact can be edited, deleted or new ones defined.

**What If?** – Allows the user to simulate the failure of a resource and generate a listing of the dependent system failures. The report generates a list of hard failures and a listing of resources saved due to redundant dependencies. The list includes type, name, role, details and contacts per resource.

**Reports** – BaconMap produces two types of reports PDF and Graph/Charts. The PDF contain detailed information while the Graphs and Charts produce summary information.

### BaconMap Reports

---

**↑ Textual Reports**

<input checked="" type="radio"/> Resource Report	Authentication Box
<input type="radio"/> Children Report	Ordered by Number of Children: Descending
<input type="radio"/> Parents Report	Ordered by Number of Parents: Descending
<input type="radio"/> Point of Contact Report	Ordered by: Last Name Ascending
<input type="radio"/> Maintenance Report	Ordered by Date: Ascending
<input type="radio"/> Resource Aging Report	Ordered by Purchase/Installation Date: Ascending
<input type="radio"/> Resource Cost Report	Ordered by: Cost Ascending
<input type="radio"/> RTO Report	Ordered by RTO: Ascending
<input type="radio"/> Role Report	

---

**↑ Graphs and Charts**

<input type="radio"/> Age Chart by	Box
<input type="radio"/> Type Cost Comparison	
<input type="radio"/> Cost Chart Comparison by	Application
<input type="radio"/> Operating Systems	<input type="radio"/> Virtual Systems

Figure 16 Reports

**Admin** – This menu option allows the user to change some components of Baconmap. The database, account and password can be changed so Baconmap can load different tables for different configurations.

**Orphan Check** – This button checks the integrity of the database. It will delete any orphaned resources. The process will provide feedback regarding the analysis.

**Group Check** – This button checks the integrity of the database for any group dependency errors.

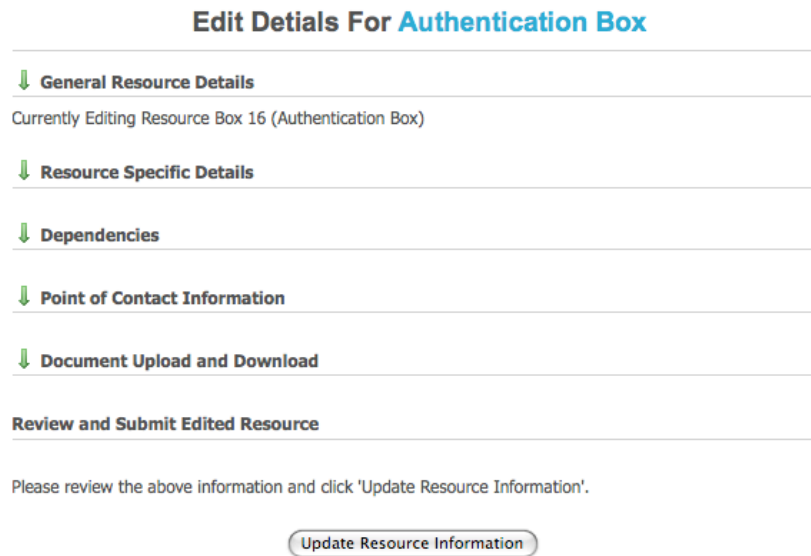
**Relationship Check** – This button checks the integrity of the database for any relationship problems.

**Full Database Verification** – Performs all three tests described above.

Users can specify what diagram shape to represent various categories of resources. In addition, roles, point of contact types, operating system and database types can be defined to suit the IT shops particular need.



## Work Panel



**Edit Details For Authentication Box**

---

↓ **General Resource Details**

---

Currently Editing Resource Box 16 (Authentication Box)

↓ **Resource Specific Details**

---

↓ **Dependencies**

---

↓ **Point of Contact Information**

---

↓ **Document Upload and Download**

---

**Review and Submit Edited Resource**

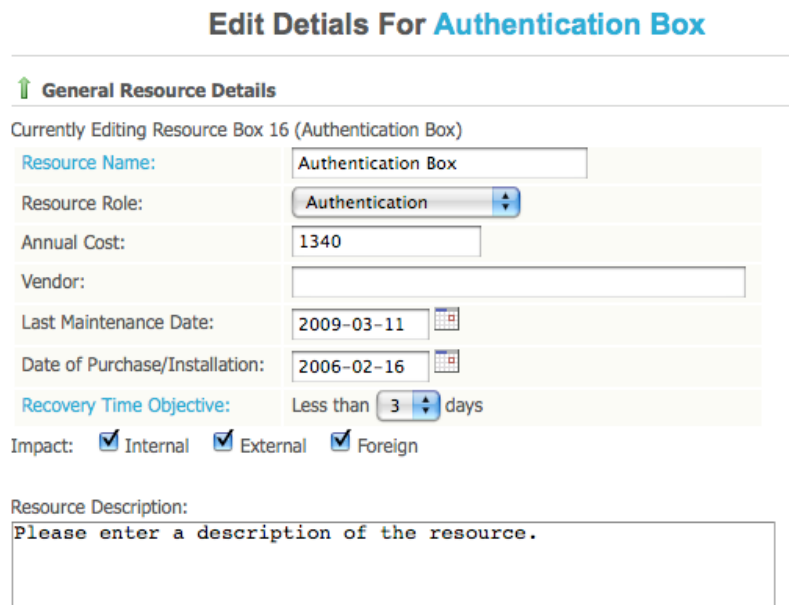
---

Please review the above information and click 'Update Resource Information'.

[Update Resource Information](#)

Figure 17 Work Panel

The work panel is the heart of Baconmap. Within this panel resources are defined and edited. This panel can be reached from the menu option or by clicking on a resource icon and selecting edit. Adding or editing a resource entry is pretty self-explanatory. The work panel consists of collapsible categories for easier readability and navigation.



**Edit Details For Authentication Box**

---

↑ **General Resource Details**

---


Currently Editing Resource Box 16 (Authentication Box)


**Resource Name:**


**Resource Role:**

**Annual Cost:**

**Vendor:**

**Last Maintenance Date:**  

**Date of Purchase/Installation:**  

**Recovery Time Objective:** Less than   days

**Impact:** ☒ Internal ☒ External ☒ Foreign

**Resource Description:**

Please enter a description of the resource.

Figure 18 Expanded Category

**Important:** the *Update Resource* Information button at the bottom of the page must be clicked on order to save the information. Here is a brief tutorial explaining the process:

**Resource name:** This must a unique name. No two resources can have the same name regardless if they are categorized as different types.

**Resource role:** This is a pulled down list that is populated from the list created by the user via the admin menu option. The resource role is typically the job the resource will fill, such as internal webserver, or DHCP service.

**Annual Cost:** This integer field is used to capture the resource maintenance and support cost. Reports can be created giving you either detailed or summary cost information.

**Vendor:** The vendor name supporting the resource is inserted here.

**Last Maintenance Date:** This date field is used to generate PDF reports listing resources and their last maintenance date. This maintenance reports can alert to upcoming maintenance or lapses in maintenance.

**Date of Purchase/Installation:** This date field is used to create resource age reports, which can be used when creating a hardware replacement policy/strategy.

**Recovery Time Objective:** This is a pull down list with integers from 0 to 30 and refers to number of days the resource can be OFFLINE. Any resource added to BaconMap defaults to 30 days and a setting of 0 indicates that the resource can have 0 downtime. RTO values are not propagated upstream from child to parents.

**Impact:** What area of your organization is impacted? Is the resource for internal use only, this is typically the IT shop. External use are areas outside of the IT shop, usually incasing the business or company. Foreign is defined as groups outside of the company. These attributes play a key role in reporting the appropriate impact scope when producing reports. More than one option can be selected.

**Resource Description:** Self explanatory.

↑ **Resource Specific Details**

---

Number of CPUs:	<input type="text" value="1"/>	CPU Speed (MHz):	<input type="text" value="2800.0"/>
Memory (Gbytes):	<input type="text" value="4.0"/>	Hard Disk Space (Gbytes):	<input type="text" value="100.0"/>
RAID:	<input type="radio"/> Yes <input checked="" type="radio"/> No	Virtual OS:	<input type="radio"/> Yes <input checked="" type="radio"/> No
Serial Number:	<input type="text"/>		
Model:	<input type="text"/>		
Physical Location:	<input type="text"/>		

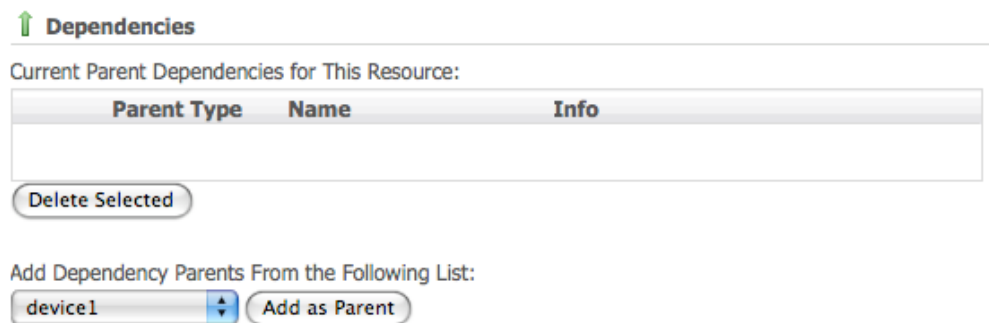
Figure 19 Resource Details

**Resource Specific Details: (Available for Box Only)** Boxes (hardware) can receive additional system information germane to computing equipment. Much of it is self explanatory; such as number of CPUs, CPU speed. The entries specific to the box resource are:

- Number of CPUs
- CPU Speed
- Memory
- Hard Disk Space
- RAID (Y/N)
- Virtual OS (Y/N)

**Resource Specific Details: (Available for Box and Device Only)**

- Serial Number
- Model
- Physical Location
- Date Acquired (Beneficial when implemented with a server replacement policy)



↑ Dependencies

Current Parent Dependencies for This Resource:

Parent Type	Name	Info
-------------	------	------

Delete Selected

Add Dependency Parents From the Following List:

device1 Add as Parent

Figure 20 Dependencies

**Dependencies:** Resources are associated with parents. The process for adding resources is always forward focused. The resource being added is always a child of an existing parent. In order to attach a child to the newly defined resource you have to select it (the newly defined resource) from the tree list on the left or add a new resource and assign the previously created resource as a parent.

An existing parent can be removed by selecting the option box and clicking on 'Delete Selected'. Baconmap will allow you to remove all parents providing you assign at least one new parent prior to updating. This is exactly how you would move an entire resource branch from one server to a different server while keeping all the children below intact.

**Point of Contact Information:** This pull down list is generated from the user created list found in the admin menu option. Edited POCs are automatically reflected on the assigned resources.

**Document upload and download:** Any type of file can be attached to the resource listed. The document should have a filename extension included to allow the browser to automatically pick the client application to use when down loading.

The uploaded file can be encrypted during the upload by clicking on the *encryption* check box prior to uploading. **IMPORTANT:** an encryption password must be set at installation time in order for this feature to work. The encryption password cannot be reset once set.

Downloading an uploaded document is done by clicking on the link or the *Download* text. If the document is encrypted you will be asked to provide the encryption password. If the password is incorrect the document will be gibberish, if it is correct the document will be legible.

Uploaded documents cannot be replaced at this time. If you wish to modify a document you must download it first and save it locally. Delete the document from the resource listing then upload the modified file.

## Appendix

eZComponents - eZ Components is an enterprise ready general purpose PHP components library used independently or together for PHP application development. [www.ezcomponents.org](http://www.ezcomponents.org)

Graphviz - Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. [www.graphviz.org](http://www.graphviz.org)

MySQL – Open source database. [www.mysql.com](http://www.mysql.com)

PHP - a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. [www.php.net](http://www.php.net)