**R**eportBuilder Server Edition makes it easy to create and deploy report applications to the web!  Build complete report server solutions for publishing reports to remote client applications and web browsers.

Using basic component configuration you can get a Windows service-based report server application up and running, a web application up and running, and have users previewing reports in a web browser in no time!
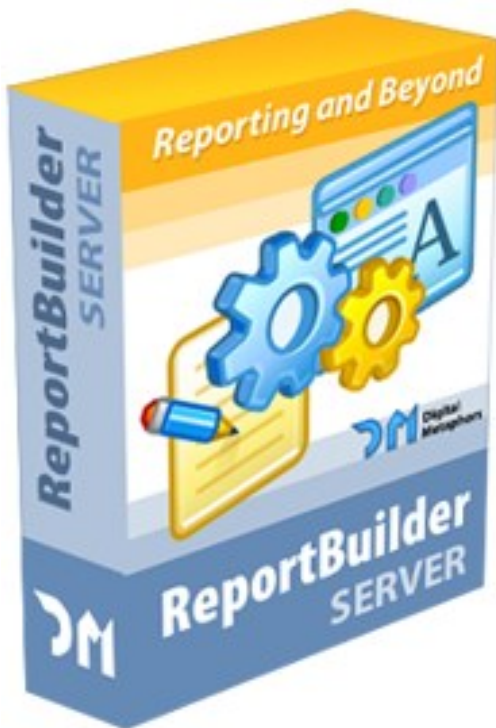
- Serve report content over the web via a multi-threaded Windows Service

- Provide a list of available reports in a folder tree structure

- Send and receive search criteria for a given report

- Publish reports as XHTML to web browser clients

- Customize the generated XHTML and JavaScript framework

- Build ISAPI, ASP, Apache web applications
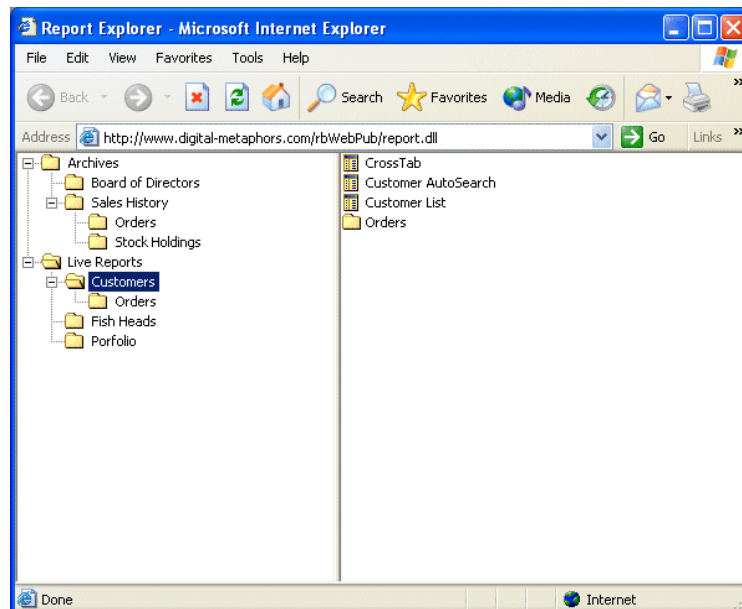
# ReportBuilder Server | Features

## ReportBuilder Services

ReportBuilder Services is a Windows service designed to act as a host to a report server application. This two-piece architecture isolates a report server application in its own process, allowing it to be started and stopped from the service. By default, the service is configured to start automatically each time the server machine boots. Once the service is started, it launches the report server application. The service is also configured to automatically restart the report server application in the event of a crash.

To install ReportBuilder Services, you simply run the included RBWinService.exe installation program. The program installs and starts the Windows service and deploys a server management application, accessible from both a system tray icon and a Start menu item. You can then access the management application and point the Windows service to the location of your report server application.

## Report Server Component

The Report Server component implements a multi-threaded report server that can execute reports and deliver report pages to clients. You build a report server application by dropping a ReportServer component on a Delphi form, creating data modules which contain ReportVolume components (explained next) and compiling the application. And that's it; you can build an entire server application without writing any code. Once the application is built you can then use ReportBuilder Services to quickly and easily designate it as the official report server.
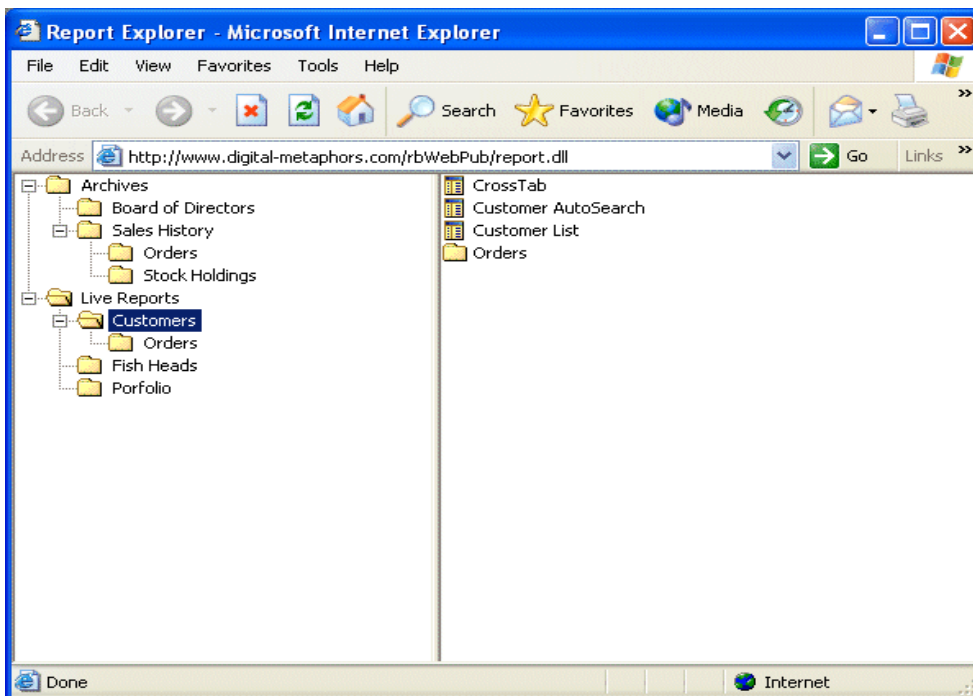
Report Explorer

The Report Server is implemented using an I/O completion port/worker thread architecture that optimally manages persistent report sessions to maximize performance and throughput. The server can generate reports incrementally (that is, page-by-page) in order to optimize response time. Pages are cached on the server and on the client to minimize network traffic. The amount of time that passes before a session expires (and the cache is cleared) can be controlled via the TimeOut property.

## WebTier Component

With one ClientReportExplorer, one ClientReport, and one line of code, you can build a full-featured Delphi reporting application which provides access to reports on a remote server. The ClientReportExplorer uses a Windows Explorer style interface to display a folder tree structure of available reports. The ClientReport makes it easy to preview and print.  When previewing a report that requires search criteria, the AutoSearch dialog is automatically displayed and any search values entered are returned to the server. The report is then generated and pages are returned.

When compiled as part of a web application, the web tier acts as a middle tier, providing access to a report server application from a web browser. A user may select reports from a folder tree structure, enter search criteria when applicable and preview report pages from the web browser. All of these capabilities are made possible via a zero-client application (100% JavaScript/XHTML) generated by the web tier. The zero-client is compatible with Netscape 4, Netscape 6, and Internet Explorer (5 and 6.)

Designed to integrate seamlessly with Delphi's various web technologies, the web tier requires only a single line of code to compile successfully as part of an ISAPI DLL or Apache Shared Module. ASP is also a snap, as a single COM interface (easily generated with the help of Delphi's ActiveX support) is all that is needed to deploy the web tier as part of an Active Server Page.

The web tier implements a sophisticated concept of "session", grouping all of the communications received from a single web browser under a single session object. In order to optimize performance, both the state of the session, and any report output generated for the session are cached. For example, assume a user previews report A, then report B, then returns again to report A. The web tier would manage all of this activity as a single session, and whenever possible, would return pages from the session cache, instead of re-requesting them from the report server application. The amount of caching which occurs, the TimeOut for a session and the frequency with which garbage collection is performed on expired sessions are just a few of the configuration options available on the web tier.

Implemented as a set of Delphi classes, the WebTier consists primarily of objects which descend from TrsWebContentProvider. Web content providers communicate with the report server application, converting the response from SOAP/XML to native Delphi objects to XHTML/JavaScript, suitable for consumption by a web browser. The content providers included in the Server Edition make it easy to provide a complete, robust, and professional web reporting solution right out of the box. You can customize the HTML created by the web tier by creating your own content provider descendants.

## ReportBuilder ISAPI Thread Pool

The Server Edition includes a custom ISAPI thread pool, implemented using an I/O completion port/worker thread architecture designed to optimally manage incoming requests. The thread pool queues requests based on session, and can eliminate redundant requests. This greatly improves both throughput and performance, as the web tier passes only relevant requests through to the report server application.

## Server Farm

The web tier can be configured to utilize any number of report servers to maximize scalability. This allows the web tier to scale gracefully as a greater number of users access reports. There are two load balancing options provided: Round Robin and Minimum Load. Round Robin simply steps through a list of servers, each time returning the next server in the list. Minimum Load checks the CPU usage of each server (based on a sixty second moving average), and returns the server with the minimum usage.

## Documentation, Help & Examples

The Server Edition Developer's Guide is provided in PDF format. The concepts behind the web server are covered, along with numerous step-by-step tutorials that guide you through everything from creating a report server application to deploying the web tier with a server farm. The online help is a reference for all of the components and major classes which make up the product. The help is integrated with the Delphi help and is accessible via the help menu or the F1 key. Numerous examples of report server applications, web applications and client applications are also provided.

## Licensing

ReportBuilder Server Edition currently includes a single developer license for ReportBuilder Enterprise and a single developer license to ReportBuilder Server. Also included is a Server Deployment license for a single CPU. This enables you to install a report server solution to a single CPU machine. Additional Server Deployment licenses are US $299.00 per CPU. Thus to deploy a server solution to a dual processor machine requires two deployment licenses. A report server can support as many clients as the hardware will allow. In summary, the licensing is per server CPU, not per client seat.

For more information, please send an e-mail to info@digital-metaphors.com.

**ReportBuilder Server Edition,** US $1,290

**ReportBuilder Server Deployment License,** US $299

**Contact:** sales@digital-metaphors.com

**Website:** www.digital-metaphors.com

**Order:** https://www.digital-metaphors.com/order