

SoftWidgets Enterprise Integration Suite (EIS) 7.10

Executive Summary

This white paper discusses how to leverage the extensive and cost effective features of the SoftWidgets Enterprise Integration Suite (EIS) 7.10. It is intended to be an architectural overview for software developers and IT decision makers who are building-out or seeking to improve their existing Enterprise Application Integration (EAI) architecture.

Contents

[Introduction](#)

[Integration the Hard Way](#)

[Principles of Easy Integration](#)

[EIS Elements](#)

[Radar](#)

[Monitor](#)

[Importer](#)

[Extractor](#)

[Uploader](#)

[UploaderD](#)

[Router](#)

[Mailer](#)

[RadarScript](#)

[SnapInHost](#)

[ExGen and EIS Extensibility](#)

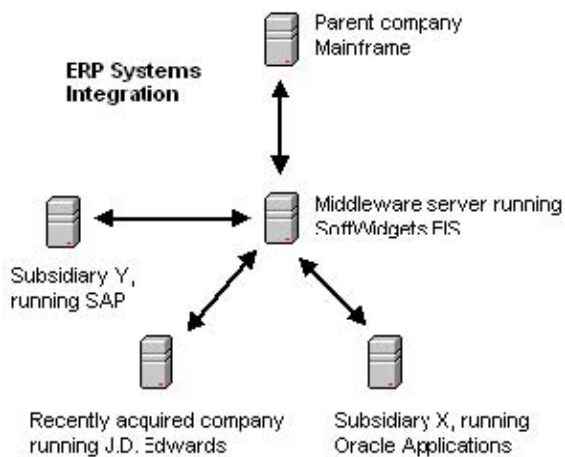
[Conclusion](#)

Introduction

The SoftWidgets Enterprise Integration Suite (EIS) is an extremely flexible, robust and cost effective way to glue together disparate systems and orchestrate processes across an entire enterprise and between business partners. When EIS is deployed, there are no bounds to the type of systems an IT department can deliver. Some typical implementations include...

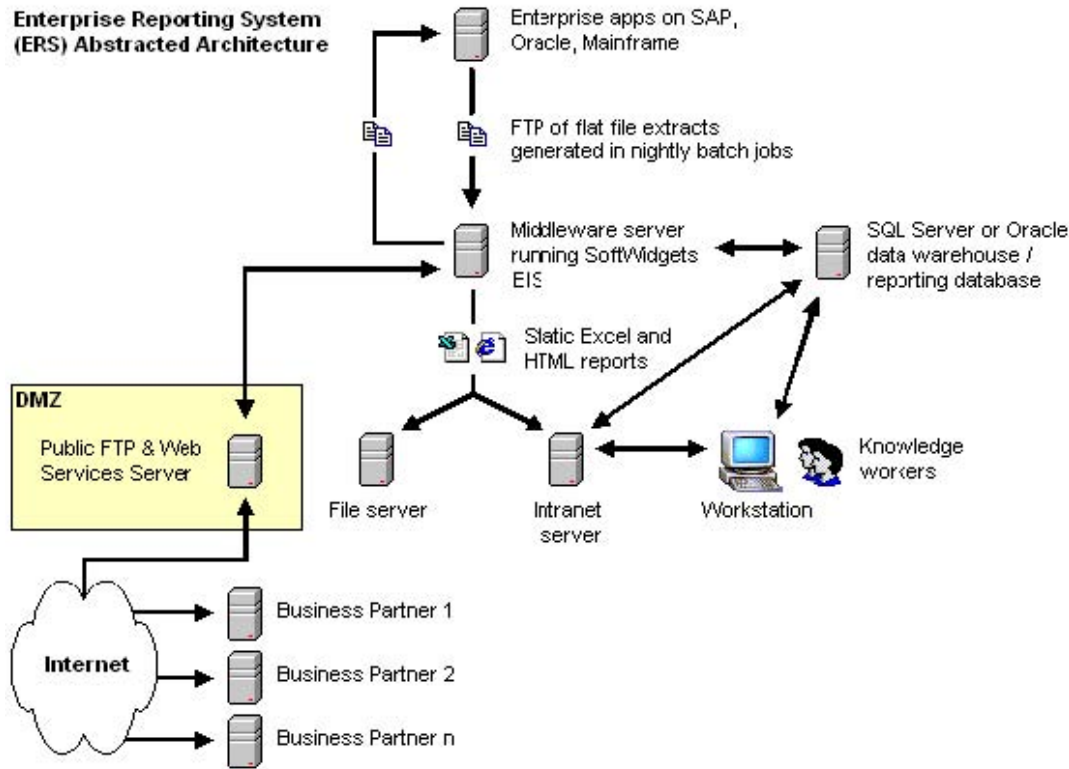
- *ERP Systems Integration*

EIS can function as the integration hub when a company needs to facilitate interaction between two or more ERP systems. In this scenario, a parent company could be running a mainframe, subsidiaries could be running SAP and Oracle applications, and a recently acquired company could be running the J.D. Edwards suite. EIS can sit in between them all, as an integration hub, and seamlessly control the flow of GL, AP and AR data that often must be shared between these systems. Below is a diagram that illustrates this scenario...



- *Enterprise Reporting Systems*

EIS is exceptional as an enterprise reporting application server. It can generate static batch reports in many different formats including simple HTML as well as the feature-rich and extremely popular Excel format. It can also manage complex database population requirements for more dynamic web-based reporting systems. Below is a diagram that illustrates this scenario...



- *Database population / data transformations*

EIS can handle the full automation of just about any Oracle, IBM DB2 or Microsoft SQL Server database population / transformation task without any custom development. Many enterprises today are required to keep several database in sync for various reasons -this is one of the real strengths of EIS.

- *Email automation*

Many applications can benefit from the advanced email automation capabilities that EIS can provide. A common example is the transformation of a manual business process into an email automated workflow.

- *File routing*

Perhaps the most simple and common use of EIS is file routing between platforms. There are literally thousands of scenarios where moving files to and from a Mainframe, a UNIX server and a Windows server is a requirement. As with many common middleware tasks, EIS can do this via simple configurations -no coding is necessary.

Building out an integration infrastructure with EIS is an excellent way to ensure that your IT department can keep up with the ever increasing demands of your organization while staying clear of the complexity-induced paralysis that is all too familiar to many large and mid-sized businesses today.

Integration the Hard Way

Today, Enterprise Application Integration (EAI) is extremely expensive and difficult. According to the Gartner Group, integration-related projects will, for the foreseeable future, continue to command the largest share of an IT department's budget. To understand why the current state of EAI is so abhorrent, the IT decision maker must understand the motivations behind integration software design and how expensive support services are baked into implementations in an effort to generate a continuous stream of

revenue for the vendor. Several current leading integrators today have brilliant integration software and services strategies that exploit the two most common elements of EAI today...

- Inflexibility
- Complexity

What are the ramifications of software that is inflexible and unnecessarily complex? Sky-rocketing costs and vendor dependence. This obviously works to the vendor's advantage, but it can end up crippling an IT department's ability to be agile and responsive to the application development needs of the organization.

Many so-called "open" integration software providers tout propriety as the culprit of ever escalating development and maintenance costs. Propriety alone, however, is not necessarily a bad thing. It is only when it is combined with complexity that it becomes an operational liability. Complexity leaves you dependent on the specialized knowledge that often is only known by the vendor and a small pool of highly paid consultants.

Principles of Easy Integration

Just as inflexibility and complexity are the root causes of expensive and difficult EAI, *flexibility* and *simplicity* are the core principles of easy integration. So how exactly does EIS allow extreme flexibility and, at the same time, reduce complexity? Below are just a few of the ways...

- *An anti rip-and-replace philosophy*

The inherently inflexible rip-and-replace proposition that other integration software providers promote is categorically rejected by SoftWidgets when there is no clear business case for such an approach. Why? Because such a strategy is often too invasive and hence fraught with risk. Also, many people in large IT organizations have worked long hours to develop systems that encapsulate a tremendous amount of knowledge that should be leveraged not reinvented just because of an emerging technological fad has swept the IT industry. The sole purpose of EIS is to allow you to take advantage of your existing systems. EIS is designed, from the ground up, to work well with all the major enterprise platforms. In fact, the more heterogeneous an environment is the better EIS becomes as an integration hub.

- *Focus on intra-platform communication standards*

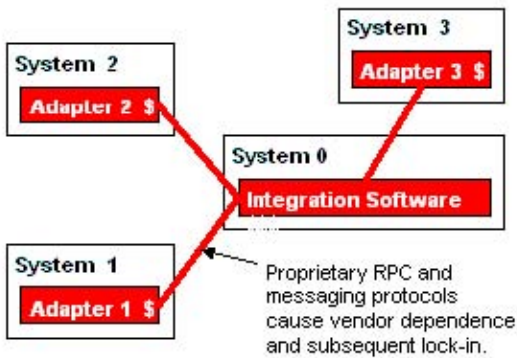
Because heterogeneity is assumed and diversity is embraced, the emphasis of EIS is on intra-platform communication standards. The big three database providers (Oracle, IBM and Microsoft) all support standard ODBC, ensuring connectivity at a database level. Virtually all platforms support FTP, most support HTTP/HTTPS and SMTP, many support LDAP, and many in the future will support emerging Web Services standards such as SOAP. The protocols just mentioned cover all the basic aspects of integration, including database access, file and content transfer, email delivery, directory manipulation, messaging, and remote procedure calls. The focus on these common denominator communication standards is the reason why EIS works very well with and leverages the strengths of virtually any platform. Acknowledgment of and respect for the relative strengths and weaknesses of a given platform is an essential tenant of EIS and it is also a key to the success of any heterogeneous integration project.

- *Noninvasive deployment*

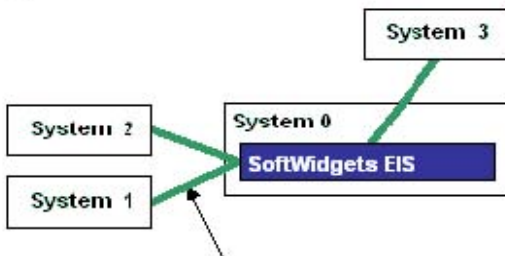
EIS is very noninvasive. In fact, the majority of EIS implementations only require a production installation of SoftWidgets software on a single computer. How can this be? How can the machines then interact? Again, it is because of the adherence to a core principle of EIS: focus on existing intra-platform communications standards, not proprietary adapter or wrapper APIs. Today, many integration packages require that you deploy their proprietary adapters or wrappers on various machines in the enterprise. This approach is inherently inflexible, disruptive to operations, often very risky to deploy and costly to maintain. It also allows the integration vendor to reinforce client dependence as pieces of their software spread throughout the enterprise. Below are two

diagrams that illustrate how the SoftWidgets philosophy is fundamentally different from other integration software providers...

Example of a costly and inflexible adapter-based integration model that serves to increase dependence and maximize revenue sources for the vendor. As seen below, this configuration requires vendor software installed on all four systems.



Example of an inexpensive and flexible integration model based on SoftWidgets EIS. As seen below, this configuration requires software installed on only one system.



NO vendor dependence or lock-in because standard intra-platform communication protocols are used. (e.g. ODBC, FTP, HTTP, HTTPS, SMTP, SNMP, SOAP)

- *Open to all / leverage existing staff*

Programmatic extension of the suite is open to all developers with knowledge of virtually any programming language. This keeps staffing costs low by allowing an organization to leverage their existing developer base and avoid the lost productivity and costs associated with retraining.

- *Built with industry-standard software and runs on commodity hardware*

The system is built with industry-standard software and it runs on commodity hardware making the infrastructure cost nearly 1/10 that of comparable systems on other platforms.

- *Mainframe-class stability*

Because of a sophisticated process isolation engine that is at the core of EIS, when it is properly deployed, it boasts "five nines" mainframe-class stability.

- *Unprecedented scalability*

The unique architecture of EIS and the concurrency management capabilities result in scalability that is on par with systems costing millions of dollars. As a result of this scalability, significant savings can be realized

through server consolidation.

- *Lego™-Like Design Philosophy*

Like a Lego kit, EIS includes several generic libraries and applications that can be reused over and over again. Combined, these applications address the majority of common integration requirements.

- *Easy operation*

Operational support requirements are minimal because of the stability and the fully automated email-based error reporting. In fact, most clients with mission-critical deployments run the suite in "lights out" mode.

The principles, outlined above, are strictly adhered to by EIS. As a result, EIS is the most robust and cost effective integration suite, in its class, on the market today.

EIS Elements

There are several elements to the EIS framework. In this section, the following key EIS applications will be discussed at a high level...

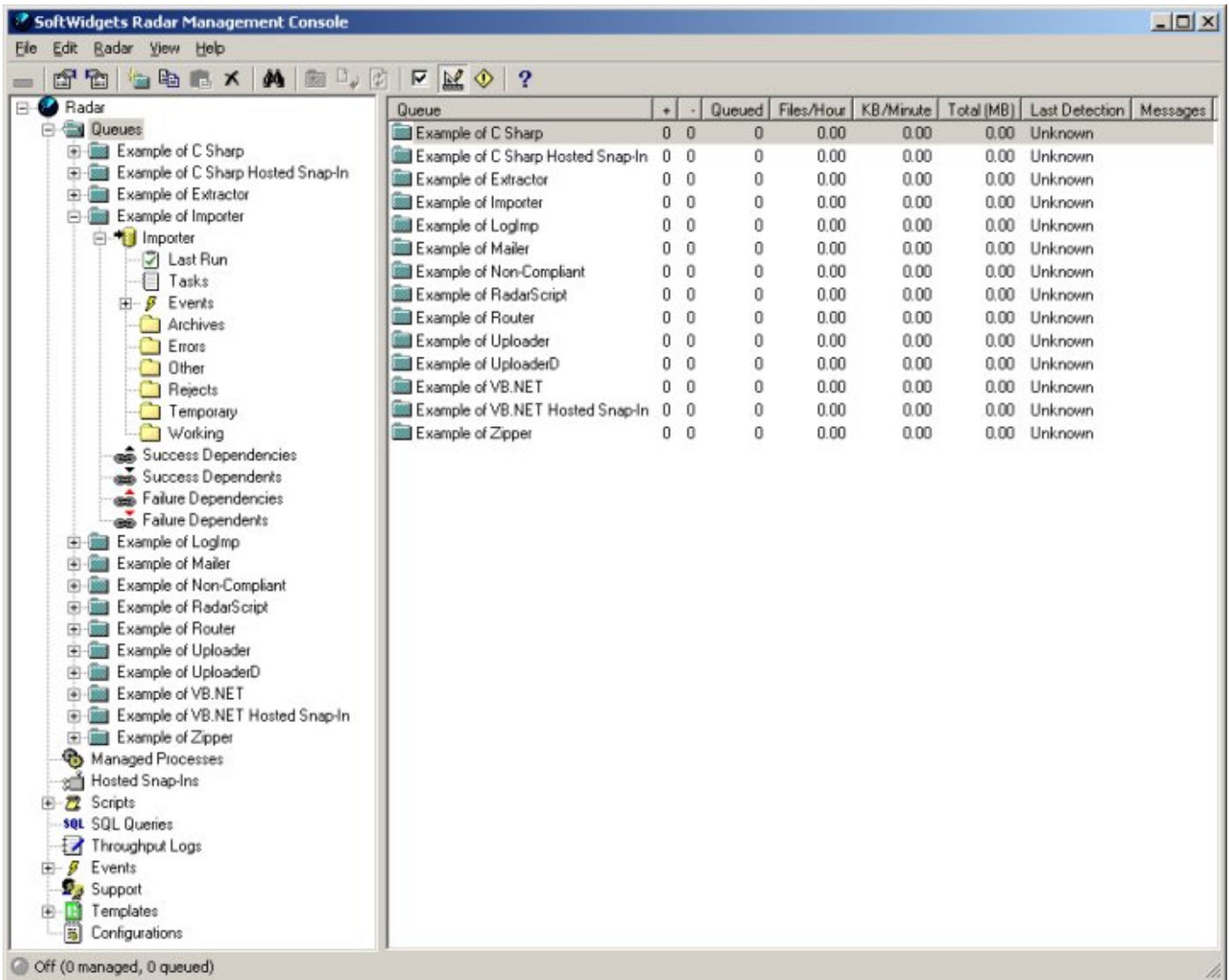
- Radar
- Monitor
- Importer
- Extractor
- Uploader
- UploaderD
- Router
- Mailer
- RadarScript
- SnapInHost

EIS Elements – Radar

What is Radar? Radar is the heart of the SoftWidgets Enterprise Integration Suite. It is a Windows service that functions as a sophisticated process manager, an enterprise file traffic cop, and the hub of a robust enterprise integration architecture. Radar can manage the execution of any process written in virtually any modern language (C/C++/C#, VB/VB.NET, Java, COBOL, Fortran, etc.). The rule of thumb is that if you can script it or compile it, it can be managed by Radar. The Radar-EIS framework also consists of .NET and COM APIs that middleware developers can program against to make an application Radar-compliant leveraging all the extensive capabilities of Radar.

At a high level, what exactly does Radar do?

- ***Watch for files to appear in queues and spawn associated processes.*** Queues are folders in your enterprise that are associated with a Radar-managed process. A good example is the common scenario where a file is extracted from a legacy system and FTPed to an integration server running Radar. When the extract file appears, Radar detects it and spawns an instance of a managed process, such as Importer, that populates a database. Below is a screen shot of the Radar UI that illustrates the relationship between a file queue and an associated managed process...

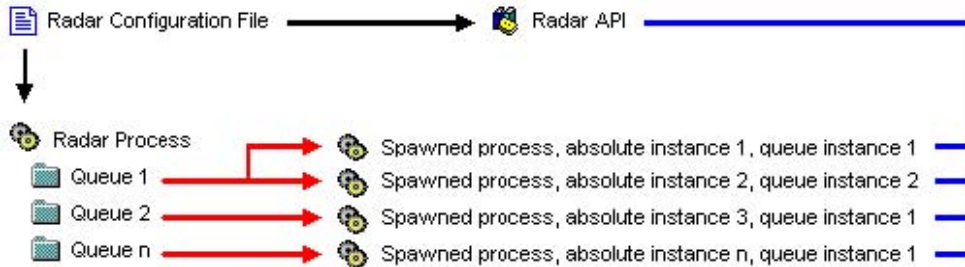


- **Controls process concurrency.** Controls process concurrency, both in a particular queue and across any number of queues. For example, if ten files are queued up simultaneously in one queue, Radar can be set to only run three processes concurrently. In this scenario, as soon as one process finishes, the next file in the queue will get processed and the window will slide up one until all ten processes are complete. This is extremely important for stability and scalability. Middleware infrastructures without this sort of process control are very vulnerable to process overload and subsequent server failure simply because there is no control over how many processes are running concurrently.
- **Chains processes.** Radar allows an administrator to chain any number of process together based on job success or failure. Chaining addresses the flexibility requirements of separate but dependent process.
- **Maintains a standard folder structure.** Perhaps one of the most important features of Radar is its ability to define and maintain a folder structure that segregates incoming files based on process and process instance.
- **Routes files.** Incoming files, upon process termination, can be routed to standard archive and error folders depending on process success or failure. They can also be routed to other queues for further processing.
- **Auto queuing.** Any queue can be configured to automatically execute jobs based on a schedule.
- **Task, event and error logging.** As part of the API, developers can leverage the powerful logging capabilities.
- **Data Throughput Metering.** Radar provides a real-time view and full metering of file traffic. Metrics include Files/Hour, KB/Minute, Total MB and more. This feature can be especially useful for business models where

revenue is a function of the amount of data processed.

With Radar in place and configured properly, you can manage thousands of individual queues and the associated processes with the piece of mind that they will all be under control.

So how does Radar manage the execution and control concurrency of any server-side process or script? The specific technical detail is the "secret sauce" of EIS and SoftWidgets and the engine that is under the hood of the Radar service is patent-pending and therefore can only be discussed at a very high level in this document. Below is an abstract diagram illustrating Radar, the queues and processes that it manages and their relationship to the EIS API...



The Radar configuration file governs the operation of both Radar and any Radar-managed process built with the Radar library. Contained in this file, is all the information for each queue that Radar manages. Any process that uses the Radar library, therefore has access to Radar's configuration. This, along with the command line conduits (red arrows) is how communication is achieved between Radar and any process that is managed by Radar.

This system design is extremely robust because there is no run-time binding between the process manager and any of the managed processes. Why does that make it so robust? Because it provides complete process isolation. In-other-words, if a managed process fails, the process manager in this architecture will NOT fail. This is in stark contrast to the more common, tightly-coupled hosted architectures that require run-time bindings via rigid interfaces.

Radar can manage multiple instances of a process at the queue level and across queues. The diagram illustrates how Radar can have four instances of a process running concurrently across three separate queues.

EIS Elements - Monitor

What is Monitor? A Windows service that is designed to track and display events reported by unattended server-based processes. Essentially, it allows an operations and development staff to scan for log files that have been written to any folder on just about any kind of server. It provides the following functionality...

- Real time event monitoring for any number of applications on any number of servers through one UI.
- Distinction / choice of three severity levels for a given event (Information, Warning, Critical)
- Ability to mitigate severity levels within an operational or "big picture" context. In-other-words, an event may be critical at a process level, but may only be a warning within the greater operational context of the enterprise.
- The ability to monitor via LAN/WAN scanning and HTTP-based scanning concurrently. HTTP scanning allows remote monitoring of any web server that supports scripting. This feature is particularly useful when you want to scan a server that is sitting behind a firewall in a DMZ.
- SMTP email integration. Any event can be configured to trigger an email message
- Robust, full featured configuration editor.
- Integrated event log viewer.
- One-button event log archive management.
- Ability to associate each event with a URL that can contain web-based reference manual for the monitored process. This feature makes troubleshooting considerably easier for non developers.
- Complete per-process log file isolation, ensuring bullet proof performance.

EIS Elements - Importer

What is Importer? A generic Radar-managed process that can import data into any ODBC-compliant data store. Importer is an extremely sophisticated application that encapsulates the majority of common middleware tasks associated with flat file database updates. The following details are handled by the Importer process...

- Connecting to databases and obtaining insertion-optimized data sets.
- Transaction management at a process and file level. In-other-words, you can import n number of files, all within one transaction or you can import each individual file as a separate and distinct transaction.
- Post process stored procedure execution. This allows for the spawning of additional processing on the destination platform.
- Pre and post importation stored procedure execution at an individual file level. This is especially handy for when an index drop is desired prior to file importation and a reindexing of the destination table is desired after importation.
- Custom transformation of data via simple and fully documented Java or Visual Basic transformation script interfaces.
- Verbose event, error and task logging.
- Post process file routing and archiving.
- Advanced thread management and process throttling.
- Diagnostics, including record count tie-outs.

Importer can handle simple fixed and delimited importation via a few configuration settings (i.e. no coding is necessary). If you have custom data transformation or scrubbing requirements then custom Java or Visual Basic transformation scripts can easily be written to accommodate those needs.

When configured properly, Importer can scale to load anything from single-record files to 1,000,000,000+ record files with carrier-grade speed, accuracy and reliability.

EIS Elements - Extractor

What is Extractor? Extractor is a Radar-managed process that allows for the extraction of data from any ODBC-compliant data source. Extractor can generate just about any flat file format that you may need with no coding required. Also, it supports rich, high performance Java and Visual Basic data transformation capabilities via a simple, standard and fully documented transformation script interface. It is the ideal tool for heterogeneous database environments in which you wish to maintain your extraction logic in the middleware tier instead of the database tier. Another key advantage Extractor has over native extraction tools that ship with most database servers is its tight integration with the rest of the SoftWidgets Enterprise Integration Suite (EIS) framework.

EIS Elements - Uploader

What is Uploader? Uploader is a Radar-compliant application that uploads files via FTP. FTP is one of the oldest computer communications protocols and it is widely supported by almost all platforms, thus making Uploader a key element of systems integration. Uploader makes any integration task that requires FTP file transmissions easy and robust by providing the following advanced enterprise-class functionality...

- Support for FTP to virtually any platform (e.g. Mainframe, Windows, UNIX and Linux)
- Guaranteed delivery via retry capability
- File name translation capability
- Ability to remotely execute or startup jobs via trigger file or stored procedure after uploads are complete
- Ability to compress and encrypt files prior to upload
- Multiple upload modes to accommodate many different transmission requirements including single, group, recursive and recursive proxy

All the features outlined above, when configured properly, provide bullet-proof FTP file transfers.

EIS Elements - UploaderD

What is UploaderD? UploaderD is a Radar-compliant application that *distributes* files via FTP to multiple destinations (hence the "D" in the UploaderD name). Among other uses, this application is ideal for distributing copies of files to a farm of web servers. At an individual destination level, UploaderD is very similar to the Uploader application. The basic difference between the two applications is the fact that UploaderD can upload files to multiple destinations.

EIS Elements - Router

Router is yet another application designed to move files around a network. It is different than Uploader, however, because it does not transfer files via FTP. It simply sends files through network shares. In general, Router requires far less configuration and network support than Uploader, but it can only be used between LDAP-compliant or Active Directory compliant platforms.

Some of the more notable Router features include...

- Full featured, graphical configuration editor
- Retry capability to ensure guaranteed delivery
- Dynamic, authenticated drive mapping capability. This is important because persistent drive mappings are not considered a best practice in server-side environments. It also allows for the case when a mapping is unavoidable because a UNC-style share reference cannot be utilized due to authentication requirements of a destination platform.

A typical Router deployment scenario would be a Windows-to-Novell and/or Windows-to-Windows intra-domain file transfer to distribute batch-generated reports to file servers.

EIS Elements - Mailer

What is Mailer? Mailer is a Radar-compliant application that is designed to automate email delivery via standard SMTP. Mailer is very simple but extremely powerful because it can be used in numerous notification and alert scenarios to streamline many business processes. Some features that make it an integration architecture building block include...

- A very simple design that leverages existing technology. To send out emails, Mailer interfaces with the standard Windows SMTP service via CDO. This design avoids reinventing the wheel and takes advantage of a proven, rock-solid SMTP mailing service.
- Guaranteed delivery.
- Ability to send file attachments.
- Extensibility and dynamic mailing capability via a simple scripting interface.

All the features outlined above, when configured properly, provide bullet-proof email delivery for any integration project. Many systems can benefit from email automation. Mailer makes it all possible without any development effort.

EIS Elements - RadarScript

What is RadarScript? RadarScript is a Radar-managed application that hosts and automates the execution of any Java or Visual Basic script. By design, Radar does not allow direct execution of scripts. The task of script execution is delegated to isolated, discrete and fully error-trapped instances of RadarScript that can be associated with Radar queues. This design insures bulletproof run-time stability and performance.

RadarScript is an ideal choice for server-side automation in three key scenarios...

- When the requirements are simple and, hence, do not warrant the implementation of a full-fledged stand-alone executable.
- When you want to host the execution of business logic that is encapsulated in a C++, VB or Java class library.
- When the code is potentially buggy.

The latter two scenarios are typical in an enterprise when the staff is overworked or not qualified to write robust Radar-compliant stand-alone applications. By directing your staff to write scripts or components that focus only on the business logic, you can have the peace of mind that no matter what errors they may have forgot to trap, RadarScript will ultimately catch them and not allow run-time errors and the subsequent hung or "dead" process that is the hallmark of un-trapped errors.

EIS Elements - SnapInHost

What is SnapInHost? SnapInHost is a Radar-managed application that can host any .NET class that implements the *Radar.ISnapIn* interface. The procedure for creating a snap-in is outlined, complete with sample code, in the *Radar* library documentation that is included with EIS. SnapInHost is similar to RadarScript in that it is an application designed to host code execution, only it is much more powerful because it allows you to write code using all the features of Visual Studio.NET and the .NET Framework.

SnapInHost is an ideal choice for server-side automation in four key scenarios...

- When your organization uses Visual Studio.NET.
- When the requirements do not warrant the implementation of a full-fledged stand-alone .NET executable.
- When you want to host the execution of business logic that is encapsulated in a C#, VB.NET or a Java class library.
- When you want the execution of potentially buggy code hosted and fully error trapped to ensure exceptional run-time stability.

SnapInHost allows for the same benefits as RadarScript when EIS is deployed in scenarios where the IT staff is overworked or not qualified to write robust Radar-compliant stand-alone applications. By directing your staff to write .NET snap-in components, you can have the peace of mind that no matter what errors they may have forgot to trap, SnapInHost will ultimately catch them.

ExGen and EIS Extensibility

An excellent example of EIS extensibility is ExGen. ExGen is a Radar-compliant Excel reporting add-on to EIS that underscores the inherent extensibility of the EIS platform. Used in conjunction with EIS, ExGen can be the foundation of a custom-tailored enterprise reporting system and/or business intelligence solution. Below are key ExGen features:

- ExGen automatically generates spreadsheets from almost any data source (Oracle, DB2, SQL Server, Sybase, Access, and more) via simple SQL statements.
- ExGen automatically routes generated spreadsheets to multiple email address and/or to network file servers.
- ExGen can populate predefined templates that have look-and-feel elements designed by business users of the spreadsheet instead of the IT staff.
- In a single run, ExGen can generate multiple workbooks.
- For each workbook generated by ExGen, multiple worksheets can be populated with data.
- Best of all, the features cited above require no custom coding! ExGen includes a very intuitive configuration interface that allows an IT professional to meet the majority of common automated spreadsheet generation and delivery requirements with no coding.
- In cases where the spreadsheet generation requirements are beyond the out-of-the-box capabilities of ExGen, extensibility is provided for via a set of fully documented .NET 1.1 compliant snap-in interfaces.

Conclusion

By combining all the EIS elements discussed above and adhering to the principles of easy integration, any IT department can be empowered to develop incredibly flexible and elegantly simple integration solutions. It is a sound and proven approach that embraces the diversity of technologies that exist within the enterprise today and maximizes the leverage that can be obtained by interfacing with existing systems. Perhaps best of all, EIS is open to virtually any developer and is by far the most affordable suite, in its class, on the market today.