

K12 Systems Integration Solution With SASIxp™

Our Sapphire Suite of products is the leading product in web based student information systems in the market today. Unlike other vendors, our solution is 100% web based. That means that every page rendered from our system is in open standards based HTML and CSS, and viewable from any system that has a valid web browser. We support multiple browsers and multiple platforms. (Internet Explorer, Mozilla Firefox, Safari, Opera, Netscape 7 to name a few, on Macintosh, PC, and Linux platforms) Reporting and complicated navigation is handled through the addition of active web content, manifested through the use of Adobe Acrobat format or Macromedia Flash format.

Client platform independence isn't our only offering... All of our software is written in Macromedia Coldfusion or pure Java code. You can run our server processes on anything that has a JVM to support it. Our goal as a company is to be flexible enough to adapt to any environment, and as such, it is an important aspect of our business model to support multiple platforms and architectures.

To that end, we also support multiple database engines for our core functionality as well as for integration. Our core database can be either Microsoft SQL Server 2000 or PostgreSQL. For larger implementations, we would recommend the former option because of the ease of administration, backups and disaster recovery, and availability of client tools to assist in most functions. PostgreSQL is a fine option as well, for those who already have an established culture of Linux across the enterprise. Our database is centralized in nature, so one server can easily support the needs of the entire district, or multiple districts, as is the case in our ASP model. It also provides archival solutions, allowing for teachers to pull up or import from any gradebook they've ever used on the system at a moments notice.

Any data access in our products needs to go through our Data Abstraction Layer. We have modules for our core functionality that implement our own backend databases. (Right now, we have one for MSSQL and one for Postgres) In addition, all interaction with a Student Information System, whether it is ours or another solution, takes place through a SMS Integration module. At present, we have integration modules for ESD Discovery™, Sapphire SIS (our product), MMS, and SASIxp™. All authentication and authorization functions are abstracted as well, allowing for integration with SQL based tables, LDAP, Active Directory, et al for user/password information, or any combination of those options.

Authorization is separate, and could happen from any of those three plus the SMS as well. In short, everything about our software that has to do with the integration with other systems has been abstracted to the level that we can drop in support for another system at a moment's notice.

It is important to stress again that the method of integration we use is *real-time* integration. When a teacher logs into our Gradebook, they are authenticated against SASIxp™ to retrieve their teacher number for the building they wish to access. It will automatically reach out to the SASIxp™ database, and pull any courses for which this teacher needs a Gradebook, at which point it will create any that need to be created. When a teacher selects a Gradebook, it will automatically reach out again and retrieve the list of students that should be present in that class, indicating any adds or drops since the last time the teacher entered the system. This allows for the most seamless integration possible from the user's perspective. Everything "just works".

The integration with SASIxp™ in particular can be accomplished via several scenarios. If the data is in MSSQL format, we can access that database directly without any issue. Coldfusion includes JDBC compliant drivers in their basic package. If the data is in a dBase IV format, we have a partner product called Sequelink which provides a high-speed dBaseIV ODBC driver and a bridge (installed on a Linux or Win32 platform) to make it work over JDBC. In that way, the flat file database used by SASIxp™ can be made available over the network, so that our web servers can get the information they need at any time. Each building could have a different back end, or not, it is completely up to the configuration of the site at hand.

For our larger SASIxp™ clients, we also provide an option that enhances performance with SASIxp™ interaction. Essentially, we create a local repository of SASIxp™ information on our MSSQL or Postgres backend, and provide a java program to keep it in sync with the backend SASIxp™ databases. This enhances performance from the user perspective, because costly operations (like saving a code into AATD or AATP with the packed fields and transaction code) that might take a few seconds to accomplish can happen in milliseconds on our backend, and then be pushed out to the slower SASIxp™ backend shortly thereafter by the java helper. Timestamps and audit trails are used to minimize data collision and make sure data is in synchronization.

Our integration with SASIxp™ is quite complete. We offer the ability to create teacher gradebooks, populate their class lists, post and receive final grades, post and view daily attendance including extended information such as tardy times, post and view period attendance, and the ability for teachers to view approved demographic information from the student, parent, and emergency atoms. We take great pride at the versatility and completeness of our integration solution in that regard.

This solution will also include our Parent Portal, which, by design, functions on its own in an autonomous fashion. It is a completely separate server and database to ensure data security. Even if the server were compromised, your data would still be safe, out of reach on the SMS/Gradebook side. All interaction with the parent portal happens in a “push” relationship, where a teacher can push reports for all their course sections to the parent portal at the same time, or report cards can be placed in the proper folders in a batch job. This ensures the best data security while making the process an expedient and convenient one.