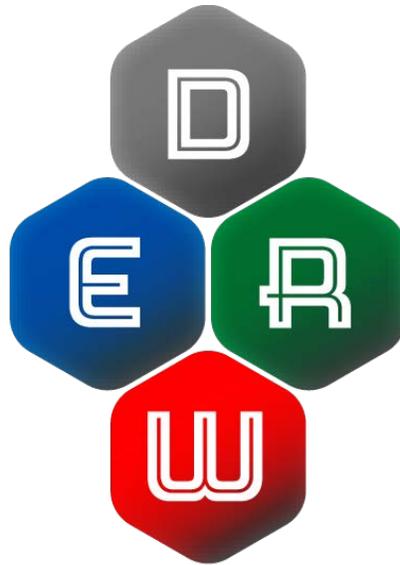


ESSENTIALS FRAMEWORK FOR DOMINO



The Problem

While the Lotus Domino server is a highly capable, standards-based platform, it is largely under-utilized and misunderstood. Data goes into Notes applications with little to no hope of getting it out, and more often than not applications are not meaningfully exposed to thin clients, existing only in the isolation of the Notes client.

Over the years, in organization after organization, Lotus Domino has been relegated to the bench in favor of other development environments, where applications are put together the 'standards-way'.

When your typical web developer, from Java/PHP, or .Net, looks at a set of application requirements, they envision streaming data in XML or JSON format, screen drawing with Javascript and Bootstrap, and styling with CSS. When Notes developers look at the same application spec, they see a confusing mess of forms, views, agents, x-pages, rich text, and Domino auto-generated content which cannot be over-written.

Over time, the following perceptions of the Lotus Domino platform have stuck: 1) It doesn't play well with others; 2) Developers on the platform need uber-specialized knowledge of the quirky ins and outs of Domino to do anything other than the basics of throwing together forms, documents, and views for an isolated audience in the Notes client.

The Solution

Quick, what's the best camera? It's the one on your phone. Why? Because 8 megapixels is actually overkill on resolution, and because it is already in your pocket, and hence available to get that spontaneous, adorable picture of your kids over to Nana.

The same can be said of Lotus Domino, and in particular the http server. It has plenty of horsepower, is already configured and waiting for use, and to continue the analogy, it's in your pocket.

Time to get Domino back into the game. The Essentials for Domino Framework is an enabler for the http stack.

Essentials instructs the Domino http server to bypass auto-generation, and do things the 'standards-way'; in doing so, it exposes among other things, database reporting and meaningful thin client instances.

Architecture

The Essentials Framework is series of complimentary modules each building on the other towards these ends:

1. Making data from Notes applications ubiquitous.
2. Empowering organizations to easily transform that data into actionable Business Intelligence.
3. Removing barriers and transforming the traditional pain points of thin client development and deployment on the Lotus platform.

The Essentials Framework resides in a single Lotus Notes application weighing in at a mere 3MB. In all respects it is a normal Notes database, which means it can be located, secured, backed-up, replicated, etc., as you would any other database in your Notes environment.

There are no plug-ins for the end user to install. There are no DLLs or other libraries required at the server level. All that is required is a Lotus Domino server with the http task running.

Essentials Framework v3.1 is comprised of three modules: Essential Data, Essential Reporting, and Essential Dev.

Essential Data



The cornerstone of any view, form, or report, is GOOD data.

Good data must be:

1. **Well-formatted.** Whether the target for that data is excel or a browser rendering, Ill-formed data, in most cases, completely cripples the application.
2. **Accurate.** Because Programming 101 says bad inputs=bad outputs.
3. **Available.** Having data and not being able to use it is the same as not having data.
4. **Targeted and Efficient.** Too much data makes a report slow to load and cumbersome to use. A lack of data, or the wrong data, makes the report incomplete and errant.
5. **Logical and Namespace Tolerant.** Often times, reports are built from multiple sources of data. Making sure one data set does not interact poorly with the other is crucial to the resultant report's integrity.

Cases in point: If I asked you to provide a current snapshot of all issues in the Tech Issues Notes database, could you? How about a snapshot from 6 months ago? Could you get our PHP developer XML from the Assets database, as they are looking to incorporate that data in a custom programming effort?

In all honesty, most of us probably answered no, or said yes, with a visible twitch and grimace. Essential Data allows you to nod yes, yes, and yes, all while sipping your tea with a steady hand.

Most databases in Notes are built as front-end centric. We focus on user-interactions to input and message data on documents. We build workflow and processing to ensure the integrity of that data. Only afterward do we stop to build some views; and always we find them inadequate and frustrating.

Essential Data allows you to take any collection of notes documents, from any database, and turn them into well-formatted, targeted XML/JSON.

Essential Data uses easily-configured notes documents called Data Profiles to describe WHERE, and HOW to get the data, as well as the FORMAT of the output data. Each Data Profile has a restful URL, which when called produces the appropriate data.

In essence, the Data Profile represents **Notes data on-demand**, as you can easily and quickly create Data Profiles representing data from 100s of sources across your Notes infrastructure.

Using XML and JSON as outputs allows the data to be consumed virtually anywhere. Whether your organization uses Lotus Domino, SQL server, Java with NoSQL, web programming with JavaScript, Excel files or Crystal Reports, Data Profiles supply the raw data in a universal, easy to use format, on demand.

Snapshots of data, which is what you get from calling URLs, represent a true step forward; however, real reporting and meaningful Business Intelligence only happens with advanced data analysis, which by definition requires comparison of multiple data sources.

The same Data Profile, which gets data restfully via a URL, can also be set up for scheduled exports. The result of the scheduled export is an XML or JSON data file. These files can be saved in meaningful directory structures on your corporate network.

Once a critical mass of data files has been achieved, real Business Intelligence begins to emerge. Using other tools in the Essentials Framework, such as Reporting and Views, users can point and configure their way to reports which compare multiple historical data sets dynamically.

Essential Reporting



Microsoft Excel has become an industry standard, and is commonly used for reporting in a majority of today's organizations.

Essential Reporting utilizes a configurable document called a 'Report Template' for displaying information from the Data Profiles in Excel.

A Report Template only requires a couple of pieces of information: 1) A link to one or more Data Profiles; 2) An attached Microsoft Excel template, which will be used to display the data.

Essential Reporting comes with a default Excel template designed to handle the basics, such as authentication to Notes, data import, normalization, and refresh. The default Excel template also includes the Essentials Pivot Table and Chart Wizard, which enables rapid, facile creation of pivot tables, pie and bar charts, all around your data.

Once Report Templates have been configured, end users can easily run reports from a simple drop down menu. Selection of a report will result in a local Excel file being created on the user's file system, with data from the selected Data Profile automatically imported and the default report drawn and delivered.

From there, the default Excel file can be modified and extended, either via the built-in Essential Pivot Table and Chart Wizard, or using normal Excel techniques. The extended Excel file can then be re-attached to the Report Template. All subsequent calls for the report will render the most recent data, and display it in the extended format.

Anyone who has worked in the reporting arena before knows well that reporting is a very iterative process. The ability to extend and re-attach an Excel template makes it yours, and keeps it relevant.

Because the Report Template is capable of having multiple specified data sources, we can begin the creation of real Business Intelligence and data analytics.

All of us have some form of Issues management database in our Notes environment, so let's use Issues as an example. We have been charged by management to deliver on-demand metrics from the Issues Management Notes application.

The Essentials Recipe Phase 1:

1. Create a Data Profile for 'Open Issues'. As the source for the Data Profile, select all documents in Open Issues view in our Issues Management database. Configure the Data Profile to include fields relevant to open issues, such as issue number, technician, priority, description, customer, category and date created.
2. Create a daily scheduled export of the Open Issues Data Profile. This will start us building historical data sets from the start.
3. Create a Report Template for 'Open Issues'. Select the Open Issues Data Profile as the source for the Report Template.
4. From the list of available reports, select 'Open Issues'. Our default report now opens in Excel.
5. Customize the 'Open Issues' Excel file as needed. In our case, we use the Pivot Wizard to create workbook pages with the following reports: Open Issues By Customer By Severity, Open Issues By Technician, Open Issues By Priority, and Open Issues By Category. We also add our corporate image to the top of each workbook page and some required corporate legalese.
6. Save and re-attach the Excel file into the Report Template.

In a number of minutes we have created an on-demand reporting capability for Open Issues.

Any time management wants Open Issues data, they simply select the 'Open Issues' report, and an Excel file opens with the most current data, stylized with corporate colors images and legalese, with workbook page reports for: Open Issues By Customer By Severity, Open Issues By Technician, Open Issues By Priority, and Open Issues By Category. They can mark up the file, forward it via email, etc., as in the end it remains a normal Excel file.

The Essentials Recipe Phase 2:

1. Repeat the same recipe as Phase 1, except we target closed issues. As such, the Data Profile for 'Closed Issues' will feature fields that relate to closed issues, such as resolution date, resolution type, and technician comments. The default Excel template will be modified to add workbook pages for reports like: Issues By Date Closed, Closed Issues By Resolution Type, Closed Issues By Customer.

Now management can look at Open or Closed issues.

The Essentials Recipe Phase 3:

1. At some point we achieve enough of a critical mass of historical data that we can begin modifying the Excel templates to show current data versus historical data, allowing management to view the current Open Issues by Category data against Open Issues By Category from last month. Concrete Business Intelligence has been achieved.

Essential Web



Implementing Essential Data and Essential Reporting has rendered data out and reporting as non-considerations. They are simply ubiquitous. Whether we are starting with a 10 year old Notes database with gigabytes of data, or starting with a new set of application requirements, getting data out and making sense of that data is an exercise in configuration and requires no development.

In a nutshell, Essential Web is about removing the barriers that make it so difficult to make meaningful use of Notes applications on thin clients.

Essential Web starts with Essential Views, which is everything a notes view should ever have been and more-or-less mirrors the View functionality incorporated in the SQL development world.

Essential Views uses easily-configured notes documents called View Profiles to describe how a thin-client should render data from selected Data Profiles. After selecting a Data Profile, users of the View Profile select which columns will be available in the rendered thin client view.

Once the View Profile has been saved and opened in a browser, any end user can easily format the view via drag and drop configuration to create something meaningful for them. These view interfaces can then be saved and shared with others. Much like extending the Excel templates in the reporting sphere, extending the View Profile iteratively builds beautiful results in no time.

Essential Views are ABLE. Right after creation, they are searchable, filterable, customizable, sortable, printable and exportable.

Returning to our issues example, here's our Phase 4 recipe:

1. Create 2 new View Profiles, one for Open and one for Closed Issues. Begin by selecting Data Profiles for each.
2. Next, select which columns from the Data Profile will be available in our view.
3. Save and open the View Profile by its URL in any browser on any device.
4. Configure one to many View Interfaces via the supplied drag and drop settings mechanisms.
5. Link the View Profile URL into some other html page so users can find and use it.

Just like data out and reporting, creating powerful web views is a non-consideration with the Essentials Framework.