

.NET Generic Report Submission and Distribution Implementation

Gordon Yan
Jan 1, 2003

Table of Contents

INTRODUCTION.....	3
REPORTING INFRASTRUCTURE DESIGN.....	4
1. Overall Architecture	4
2. Interface Module	6
3. Report Server.....	7
4. Distribution Scheduler.....	8
5. Database.....	9

Introduction

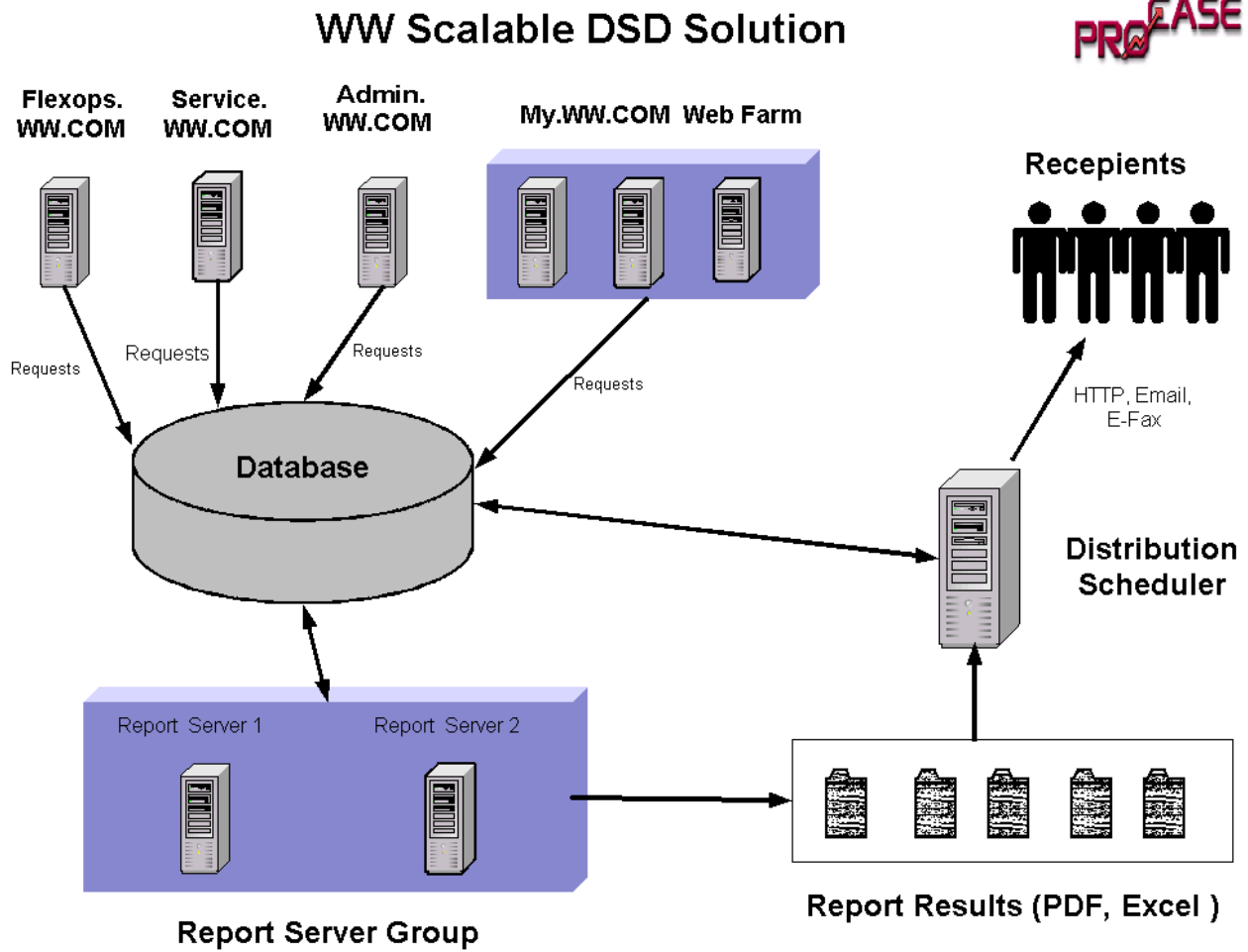
This reporting infrastructure Procase developed and deployed for a client is a common report scheduling and distribution foundation for all client's web applications. The first version of this infrastructure was released in September 2002.

This document provides technical design information about this reporting infrastructure.

Reporting Infrastructure Design

1. Overall Architecture

The reporting infrastructure has 4 tiers: interface module, report server, distribution scheduler, database. The following picture shows the overall design:



Client has multiple applications which need to use reporting infrastructure functions. "Multiple Application Shared" concept is taking into design consideration for all 4 tiers design, only the interface tier needs to be customized for each application.

Interface module is the applications that submit report job requests, create distribution sets, check report and distribution set status. Each application has its own interface pages to access reporting infrastructure functions. They have similar functions, but "Look and Feeling" is customized for each application.

Report server is the module that processes report requests sent from all applications and produces report files in PDF or Excel. Load balance has been built into report server design to handle intensive requests sent from many applications.

Distribution scheduler is the module that processes distribution requests sent from all applications and sends produced reports to recipients as HTTP links in email based on scheduled time of distribution set.

Database stores all report configuration data, report and distribution request queues. It also provides stored procedure packages to support interface module and report server.

2. Interface Module

Interface module is the place where reporting infrastructure functions are exposed to end users. It includes ASP pages and middle tier component to support these pages.

EASY CUSTOMIZATION

Client has multiple applications which need to use reporting infrastructure functions and each of them has different interface styles. To provide user friendly interface for each application, the interface module is customized for each applications and the reporting features will appear in the “Right place with right look and feeling” in each application. For saving customization effort, the interface module is designed for easy customization. A single independent DLL is designed to provide middle tier support for the whole interface module. Several include and ASP pages can be easily plugged into different web sites and customized for them.

SECURITY

Security is designed into interface module. When a user login to an application, he/she can only access reports defined for his/her group for this application. He/she can only see report jobs and distribution sets created by himself/herself in this application.

DATA DRIVEN REPORT PARAMETER PAGE

Report parameter page is a completely data driven page. When a new report is properly added to report configuration tables, a corresponding new report parameter page will appear in the report submission screen automatically without designing a new report parameter page. This will greatly reduce maintenance effort.

VIEW REPORT OUTPUT ONLINE OR SEND REPORT OUTPUTS TO RECIPIENTS

A user can submit a standalone report quest and view the report output through the web interface, or he can create a distribution set and send report outputs to multiple recipients in an email. A recurring distribution set will send updated reports to recipients periodically (daily, weekly, monthly) without a user further attention.

3. Report Server

Report server is the module to process report requests and produce report output files. It's invisible for the end users and installed on a separate machine. It will be started by Windows schedule. Once it's started, it will run continuously as a service.

CRYSTAL REPORT BASED REPORT SERVER

Report server is a Crystal Report based design. It checks report request queue periodically. If a pending report request is found, it will change this request status, find the proper Crystal Report template and create an instance for it, call related stored procedure and bind it to the report instance to produce the report output file and save it to file system.

PDF AND EXCEL FORMAT

The report output format can be either PDF or Excel. PDF is for nice report format. Excel report can be used for data analysis.

LOAD BALANCE

As report server will process requests for all applications, scalability is important for it. To address this issue, load balance is built into report server design and multiple report servers can be easily configured to run in parallel to reduce response time.

EXECUTE REPORT JOB ON SCHEDULED DATE AND TIME

If a report job is not attached to any distribution set, it will be executed as soon as report server gets time to process it. If a report job is attached to a distribution set, this report job will be executed on the scheduled date and time of the distribution set.

4. Distribution Scheduler

Distribution scheduler is the module to send report outputs as Http link in email to recipients based scheduled time of distribution set. It will be installed on the same machine as report server. It will be started by a Windows schedule periodically.

PROCESS ONE TIME DISTRIBUTION SET

It checks distribution requests periodically. If a pending distribution set is found, it changes the distribution set status, find all report jobs and recipients related to this distribution set, form an email embodied with Http links pointing to report output files and send it to all recipients.

PROCESS RECURRING DISTRIBUTION SET

A user can create a recurring distribution set which will send updated reports to recipients periodically and automatically. When distribution scheduler finishes a recurring distribution set, it will duplicate this distribution set using the same recipients and report parameters and set its due date to the next scheduled date.

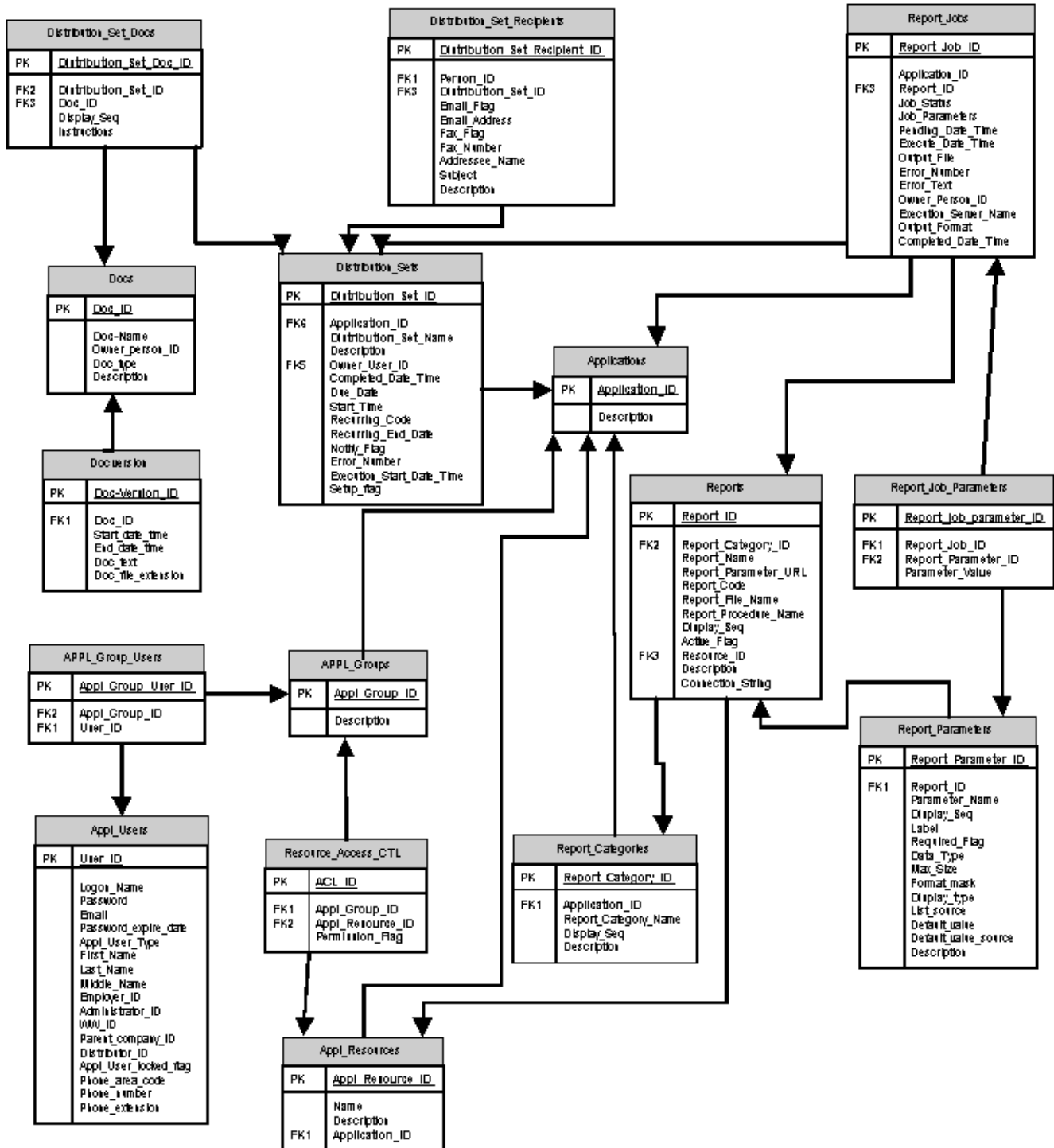
EFAX SUPPORT IN THE FUTURE

When client gets an EFax service account, this module will be improved to support sending report outputs to recipients via fax.

5. Database

Database store all report configuration data and report request queue and distribution request queue. All other three tiers communicate with database for submitting requests, check request status, process requests and distribute reports to recipients.

The following diagram describes reporting infrastructure data model.



MULTIPLE APPLICATION SHARED REPORTING TABLES

In data model design, one key consideration is “Multiple Application Shared” concept. All applications will share the same set of reporting infrastructure tables. Application ID is introduced in several key reporting tables for this purpose.

SECURITY TABLES

Client’s security tables, like “Appl_users”, “Appl_groups”, “Appl_group_users”, “Appl_resources” and “Resource_access_CTL” are used for reporting access control.

Mainly interface module deal with these tables.

REPORT CONFIGURATION TABLES

Tables “Report_categories”, “Reports” and “Report_parameters” are report configuration tables. Table “Report_parameters” is where we define new report parameter pages.

Fields “Report_File_Name” and “Report_Procedure_Name” in “Reports” table are introduced to define Crystal Report template and related stored procedure for each report. Field “Report_code” is kept for Oracle Report Server. Field “Resource_ID” in “Reports” is for access control.

Mainly interface module deal with these tables.

REPORT REQUEST QUEUE TABLES

Tables “Report_jobs” and “Report_job_parameters” define report requests. Table “Report_jobs” is the key report request queue table.

Fields “Job_status” and “Execution_server_name” in “Report_jobs” are for report server load balance. Field “Output_format” in “Report_jobs” is for report format “PDF” or “Excel”.

Mainly interface module and report server deal with these tables.

DISTRIBUTION REQUEST QUEUE TABLES

Tables “Distribution_sets”, “Distribution_set_recipients”, “Distribution_set_docs” and “Report_jobs” define distribution requests. Table “Distribution_sets” is the key distribution request queue table.

Field “Recurring_code” in “Distribution_sets” is for recurring process.

Mainly interface module and distribution scheduler deal with these tables.

DOCUMENT MANAGEMENT TABLES

Tables “Docs”, “Doc_Versions”, “Distribution_set_docs” are included for future document management functions.