Outlook Calendar 2 overview

The new Outlook Calendar Service (v2) has been rewritten from the ground up in C#; although most of the logic of the application has been kept, the main change is that the solution does not use the Redemption third party library to talk to Exchange but instead uses Exchange’s managed API to communicate with it

To make development and testing of the application easier, the solution contains 4 projects

# OutlookCalendarLogic project (Class library)

This class library contains all the logic required to actually retrieve the work to be done, process it and then talk to Exchange to create calendar entries as required

Having a separate project handling the logic has the benefit of behaving as an API that other projects (as described below) can use; this way we can have a Windows service application as well as a Windows Forms application using and testing the same logic

## The “Worker” class

This is the main class of this project and it’s where most of the work takes place

This class raises message events as part of its operation; some examples of messages are “Could not logon to Exchange” or “Created calendar entry”; these messages have an associated severity (i.e. “Error”, “Information”) and event type (“Windows Event Log”, “Debug Log”); applications that use this library can add their own message event handlers and then process the messages as needed (.i.e. write them to a file, to the Windows Event log, to a text box)

The two main public methods of this class are (briefly) described below.

* GetAndCheckUserConfiguration. Get the user-defined values from the config file and make sure they are valid
* ProcessEntries. Call the private method OutlookBatch, which carries out most of the work (see below)

Some of the other public methods include:

* OutputUserConfiguration. Do as the method name says
* DescribeServicedSystems. Outputs the SQL Server / Databases that are being serviced by the application
* CreateTestCalendarEntry. Do as the method name says

Some of the private methods include:

* LogonToExchange. Attempt to login to Microsoft Exchange with the credentials provided, which can be either provided explicitly or obtained from the user running the application
* Outlookbatch. Fetch outlook events from the database, process them and create or delete calendar entries as needed
* DatabaseIsOk, DatabaseIsLocked, DatabaseIsBeingServiced, DatabaseIsRunningOvernight, DatabaseVersionIsOK. Methods to do several checks on the database
* CreateEntry. Create an appointment in Outlook
* DeleteEntry. Delete an appointment in Outlook

# OutlookCalendarService project (Windows Service)

This is the Windows Service that uses the OutlookCalendarLogic library to do the work

The service works by doing the following

1. Gets the email address under which the service is running; this is to be used as the credentials to login to Exchange
2. Creates a new instance of the Worker class using the value from the previous step
3. Sets up message event handler to capture the messages spawned by the Worker class; as the messages arrive, they are either logged to Windows Events or written to a diagnostic file (if configured)
4. Reads the user configuration file; if any errors are detected, the service stops
5. Outputs the user configuration and the serviced systems
6. Sets a timer so it can process the outlook entries periodically
7. Starts the timer

The timer triggers the ProcessEntries method of the Worker object, which does the work as described in the previous section

# OutlookCalendarServiceTester project (Windows Forms)

This application is a fat client that allows either Support or the end user to test their connectivity to Exchange; it uses the OutlookCalendarLogic library as well; during development/debugging of the main library (OutlookCalendarLogic), this application is very useful to test the main logic, since trying to test the logic using a Windows Service is notoriously more difficult

The end user and Support can use the application to create a test entry in their calendar and get diagnostic information from Exchange

Support can also use the application to run the ProcessEntries method once, test the auto discover feature of Exchange and check the user configuration

# OutlookCalendarService2Setup project (Windows installer)

This project installs the Windows Service