

3D Inspection System 12 API information may be used to import or export information to or from his own application into version 12 Office Management using database models and universal controllers to interface with BOTH local running and future cloud-stored versions of our database, while maintaining complete control over one's own interface coding.

This option provides the most flexibility to tie into both our future cloud stored database option and local stored versions at the same time, with the same interface, and do virtually anything you want with the data.

An additional intermediary control is also available to make it easier for a third-party scheduler program to send their information to 3D. Below is a sample of code that might be used with the intermediary piece designed for that purpose (which could be used in place of or along with the other API information):

Here is a sample:

```
private void button11_Click(object sender, EventArgs e)
{
    // Create some data
    var jobs = new List<JobModel>();
    var job = new JobModel()
    {
        ExternalLinkID = "11335",
        Active = true,
        Contact = "Charles",
        Directions = "Turn left anywhere",
        Duration = 3,
        InspectionDate = new DateTime(2013, 10, 1, 15, 0, 0),
        Email = "charles@gmail.com",
        Instructions = "Don't get blood on the floor.",
        Notes = "These are notes. These are only notes. I didn't want to think about notes.",
        ShipAddress = "554 Cleveland Ave",
        ShipCity = "Cape Coral",
        ShipState = "FL",
        ShipPostalCode = "33901",
        Occupant = "John Smith",
        ShipName = "John Smith",
        SiteAddress = "552 Cleveland",
        SiteCity = "Cape Coral",
        SiteState = "FL",
        SitePostalCode = "33903"
    };
    // Set an inspector on the job
    job.Inspector.ExternalLinkID = "4449";
    job.Inspector.Name = "Billy Johnson";
    job.Inspector.Address = "3498 Cleveland";
    job.Inspector.City = "West Palm";
    job.Inspector.State = "FL";
    job.Inspector.PostalCode = "33912";
    // Set contact 1
    // NOTE - figuring out which contacts should be associated could be done by looking at
    // Supp3DUI.StaticData.Preferences.Contact1Name - Contact7Name OR by using an option of some
    // kind.
    job.Contact1.Contact.ExternalLinkID = "5001";
    job.Contact1.Contact.Name = "Heath, Toby";
    job.Contact1.Contact.FirstName = "Toby";
    job.Contact1.Contact.LastName = "Heath";
    job.Contact1.Contact.Address = "4509 S. Terry Street";
    job.Contact1.Contact.City = "Cape Coral";
    job.Contact1.Contact.State = "FL";
    job.Contact1.Contact.PostalCode = "33901";
    // Add a line item
    job.Invoice.Add(new JobInvoiceModel() { ExternalLinkID = "5443", Amount = 200, Description =
```

```
"Do an inspection", Name = "INSPECT", Price = 200, Quantity = 1, Visible = true});
jobs.Add(job);
// Import the data
var m = new Supp3DUI.ViewModels.ImportScheduleViewModel(jobs, Supp3DUI.StaticData.Preferences);
Supp3DUI.MvvmFramework.Windows.WindowsDispatcher dispatcher = new Supp3DUI.MvvmFramework.Windows.WindowsDispatcher(null);
dispatcher.ShowDialog(m);
}
```