Overview

Integral Enterprise DataBridge (IED) gives users much data integration, transformation and delivery capability. Below is an example of this integration with Mapics SyteLine version 7.03 ERP.

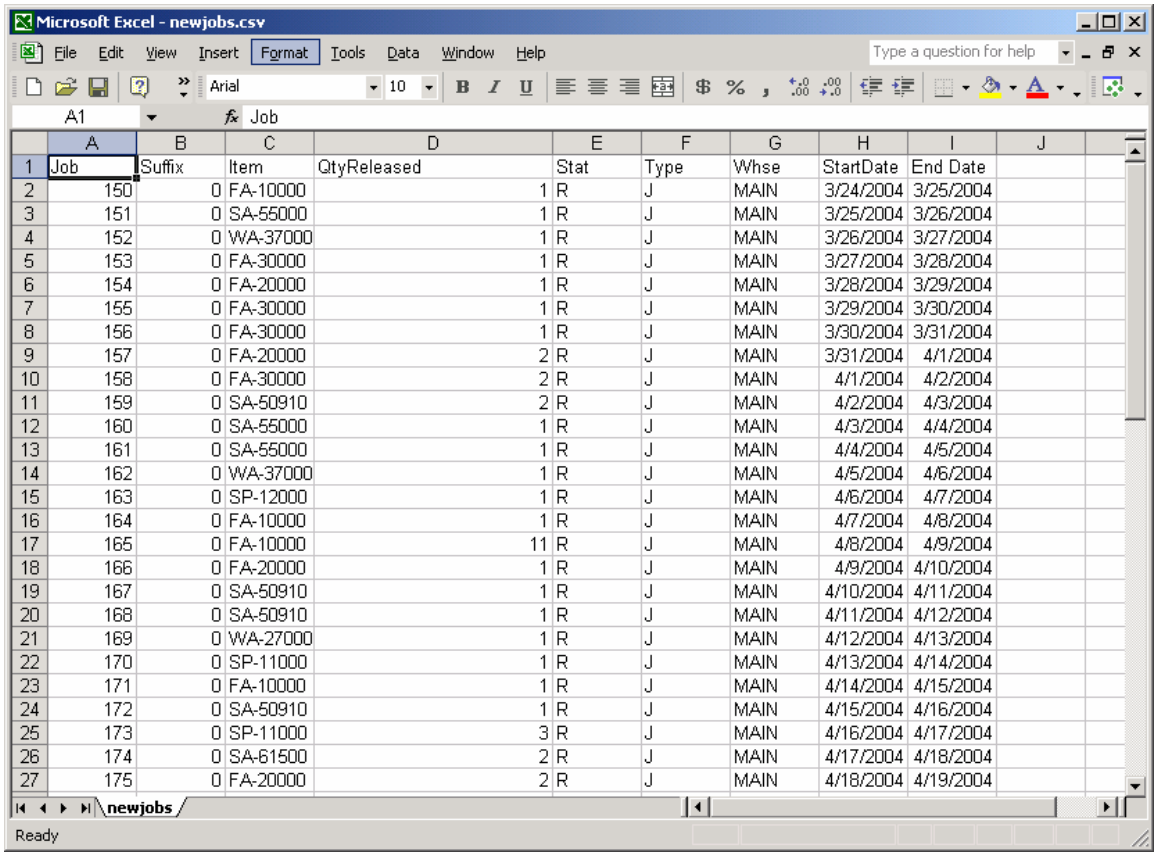
In this example, we are using SyteLine's "IDO's" – a published set of COM objects that allows data retrieval and manipulation – to load Job Orders from a comma-delimited ASCII file into SyteLine. This allows the use of existing SyteLine business rules to validate the data as it's going in.

The Job Orders are loaded with a status of Released. Also, the current Item BOM is copied into the newly created job for the given item, so that operations and material structures are also loaded.

Of course, IED can retrieve data from many different sources, including Integral Enterprise Viewer (IEV) views, from a Microsoft SQL ® or Progress ® RDBMS database, or from an XML file. Data can also be transported to different systems via the Internet as well – securely yet easily, and be delivered directly to a database system, written as flat-files, or loaded via COM objects as in this example.

In order for this example to work, you will need a flat file very similar to the one used in the example, and the IED Service (a Windows service that actually does the work of data transformation and delivery) will need to be installed on a server that also have SyteLine 7 installed on it as well.

First, let's take a look at the sample job text file, as seen in Microsoft Excel ®. The file is simply called "newjobs.csv" located on the root of my C: drive (this is an example, after all).



Microsoft Excel - newjobs.csv

File Edit View Insert Format Tools Data Window Help — Type a question for help

Arial 10 B I U

A1 — Job

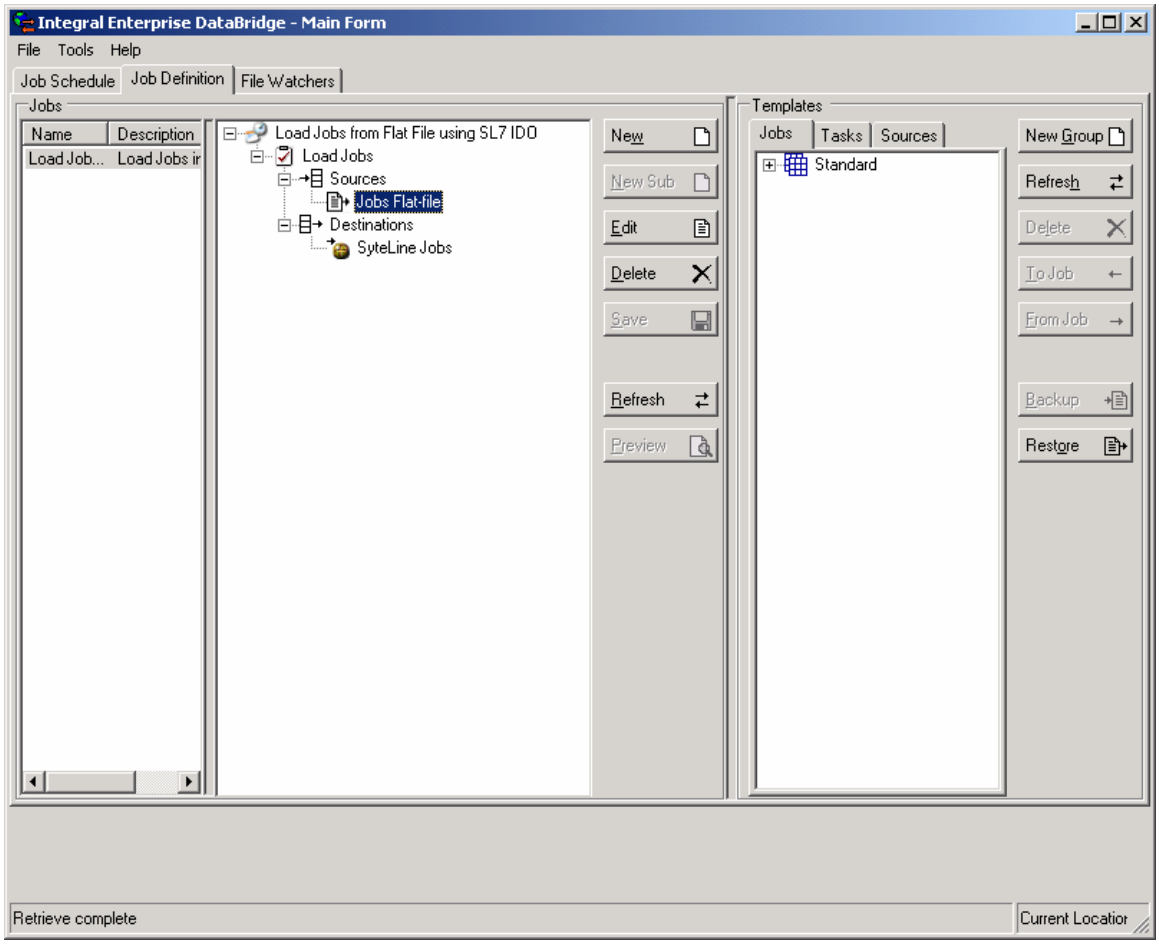
	A	B	C	D	E	F	G	H	I	J
1	Job	Suffix	Item	QtyReleased	Stat	Type	Whse	StartDate	End Date	
2	150	0	FA-10000	1	R	J	MAIN	3/24/2004	3/25/2004	
3	151	0	SA-55000	1	R	J	MAIN	3/25/2004	3/26/2004	
4	152	0	WA-37000	1	R	J	MAIN	3/26/2004	3/27/2004	
5	153	0	FA-30000	1	R	J	MAIN	3/27/2004	3/28/2004	
6	154	0	FA-20000	1	R	J	MAIN	3/28/2004	3/29/2004	
7	155	0	FA-30000	1	R	J	MAIN	3/29/2004	3/30/2004	
8	156	0	FA-30000	1	R	J	MAIN	3/30/2004	3/31/2004	
9	157	0	FA-20000	2	R	J	MAIN	3/31/2004	4/1/2004	
10	158	0	FA-30000	2	R	J	MAIN	4/1/2004	4/2/2004	
11	159	0	SA-50910	2	R	J	MAIN	4/2/2004	4/3/2004	
12	160	0	SA-55000	1	R	J	MAIN	4/3/2004	4/4/2004	
13	161	0	SA-55000	1	R	J	MAIN	4/4/2004	4/5/2004	
14	162	0	WA-37000	1	R	J	MAIN	4/5/2004	4/6/2004	
15	163	0	SP-12000	1	R	J	MAIN	4/6/2004	4/7/2004	
16	164	0	FA-10000	1	R	J	MAIN	4/7/2004	4/8/2004	
17	165	0	FA-10000	11	R	J	MAIN	4/8/2004	4/9/2004	
18	166	0	FA-20000	1	R	J	MAIN	4/9/2004	4/10/2004	
19	167	0	SA-50910	1	R	J	MAIN	4/10/2004	4/11/2004	
20	168	0	SA-50910	1	R	J	MAIN	4/11/2004	4/12/2004	
21	169	0	WA-27000	1	R	J	MAIN	4/12/2004	4/13/2004	
22	170	0	SP-11000	1	R	J	MAIN	4/13/2004	4/14/2004	
23	171	0	FA-10000	1	R	J	MAIN	4/14/2004	4/15/2004	
24	172	0	SA-50910	1	R	J	MAIN	4/15/2004	4/16/2004	
25	173	0	SP-11000	3	R	J	MAIN	4/16/2004	4/17/2004	
26	174	0	SA-61500	2	R	J	MAIN	4/17/2004	4/18/2004	
27	175	0	FA-20000	2	R	J	MAIN	4/18/2004	4/19/2004	

newjobs

Ready

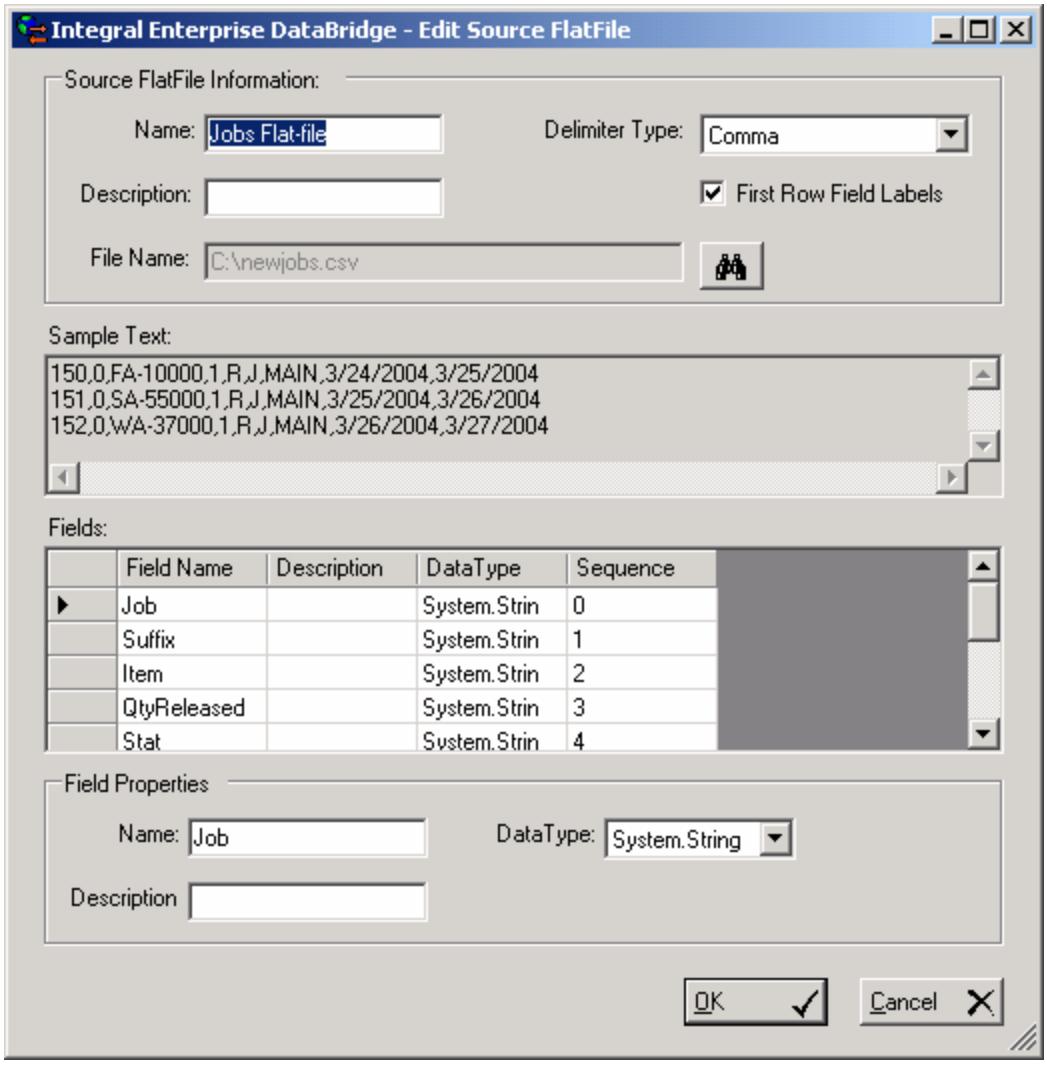
As you can see, the basic job information is there – the Job number, the Item, Quantity Released, etc.

Now, all I have to do in IED is set up a new Job, and define a source object of type “Flat File” that points to the .csv file, and a destination object of type “COM API” , and define the script needed to connect to and use the SyteLine 7 IDO library.



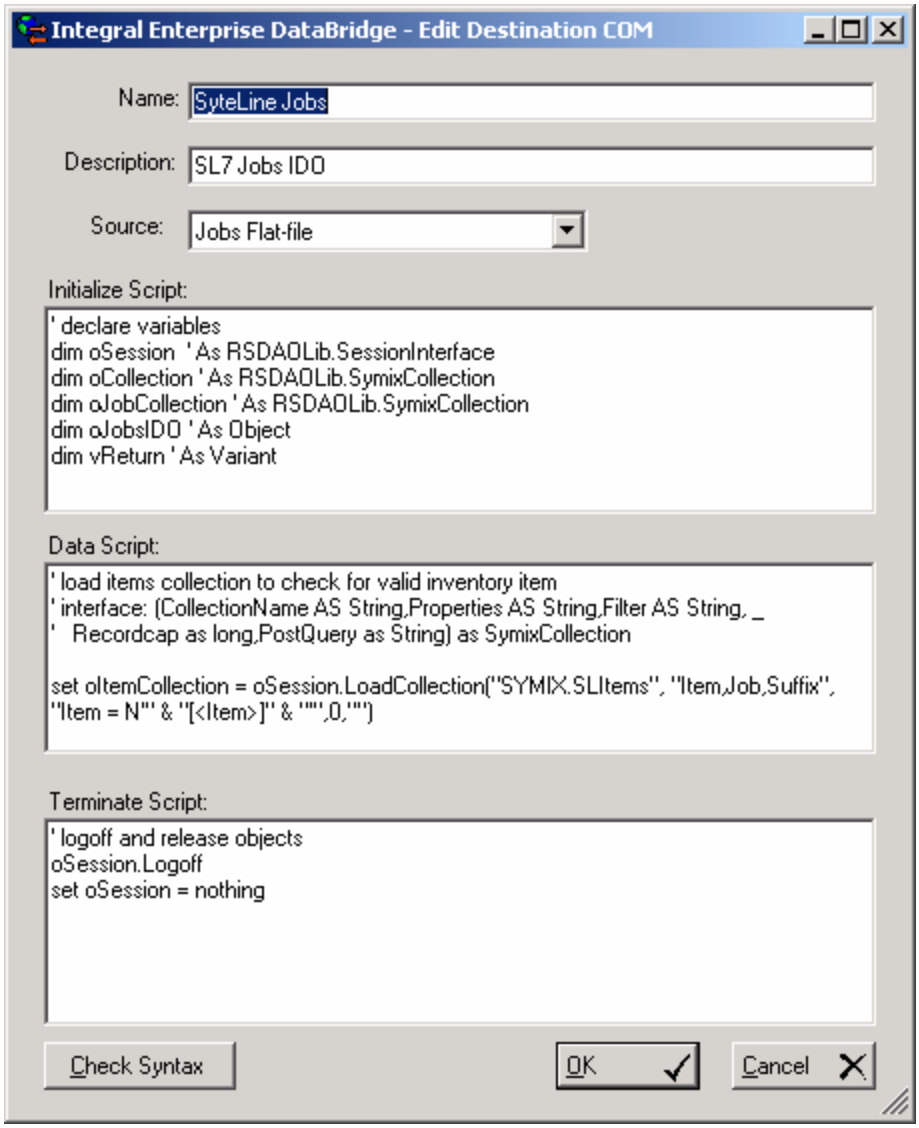
This is the job structure that I will end up with. The following details the steps of the process of creating this structure.

1) Create a new IED Job and call it something distinctive.
2) Create a new Task by clicking on the new Job and selecting “New Sub”.
3) Create a new Source Object by clicking on the new task and again selecting “New Sub”. Select “Flat File” from the pop-up menu.
4) Click on the new Source Object and click “Edit”. Set the options for the object as shown in the following screen-shot:



Notice that since “First Row Field Labels” is selected, the field names are pulled directly from the flat file. Press OK to store these changes.

5) Next, create a new Destination object by clicking on “Destinations” and then “New Sub”.
6) Set the properties as follows on the following screen-shot:



Here are the full scripts used in the example:

Initialize Script

```
' declare variables
dim oSession  ' As RSDAOLib.SessionInterface
dim oCollection ' As RSDAOLib.SymixCollection
dim oJobCollection ' As RSDAOLib.SymixCollection
dim oJobsIDO ' As Object
dim vReturn ' As Variant


' create session and logon
Set oSession = CreateObject("SYMIX.SessionInterface")
oSession.Logon "sa", "sa", "SyteLine" ' username, password, configuration
```

```
' create job collection
Set oJobCollection = oSession.CreateCollection
oJobCollection.CollectionName = "SYMIX.SLJobs"

' create Jobs IDO instance
Set oJobsIDO = oSession.CreateInstance("SYMIX.SLJobs")
```

Data Script

```
' load items collection to check for valid inventory item
' interface: (CollectionName AS String,Properties AS String,Filter AS String, _
'   Recordcap as long,PostQuery as String) as SymixCollection

set oItemCollection = oSession.LoadCollection("SYMIX.SLItems",
"Item,Job,Suffix", "Item = N'" & "[<Item>]" & "'",0,"")
if oItemCollection.Count = 0 then err.raise 1,"SyteLine Jobs","Item not found"

With oJobCollection
   .AddNew True
   .SetProperty "Job", "[<Job>]"
   .SetProperty "Suffix", "[<Suffix>]"
   .SetProperty "Item", "[<Item>]"
   .SetProperty "QtyReleased", "[<QtyReleased>]"
   .SetProperty "Stat", "[<Stat>]"
   .SetProperty "Type", "[<Type>]"
   .SetProperty "Whse", "[<Whse>]"
   .SetProperty "JobDate", "[<StartDate>]"

    oSession.BeginTrans
    .SaveChanges True
    oSession.CommitTrans

End With

' load job-sch collection to update MRP dates
Set oCollection = oSession.LoadCollection("SYMIX.SLJobSchs",
"EndDate,Job,Suffix,StartDate",  "Job = N'" & "[<Job>]" & "' AND Suffix = " &
"[<Suffix>]", 0, "")
If oCollection.Count = 0 Then err.raise 1,"SyteLine Jobs","JobSch not found"

oCollection.SetProperty "StartDate", "[<StartDate>]"
oCollection.SetProperty "EndDate", "[<End Date>]"

oSession.BeginTrans
```

```
oCollection.SaveChanges True
oSession.CommitTrans

oJobsIDO.JobOrdersCopyJobSP "[<Job>]", "[<Suffix>]", "[<Item>]", "", vReturn
```

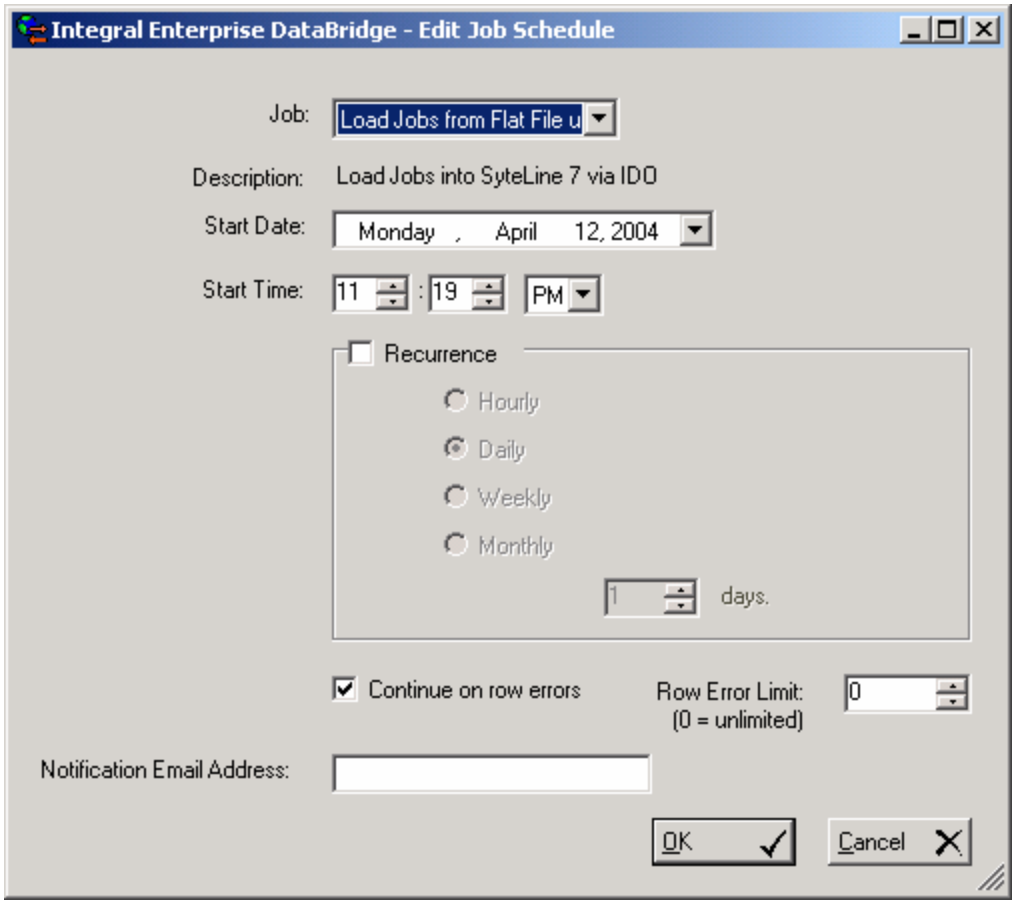
Terminate Script

```
' logoff and release objects
oSession.Logoff
set oSession = nothing
```

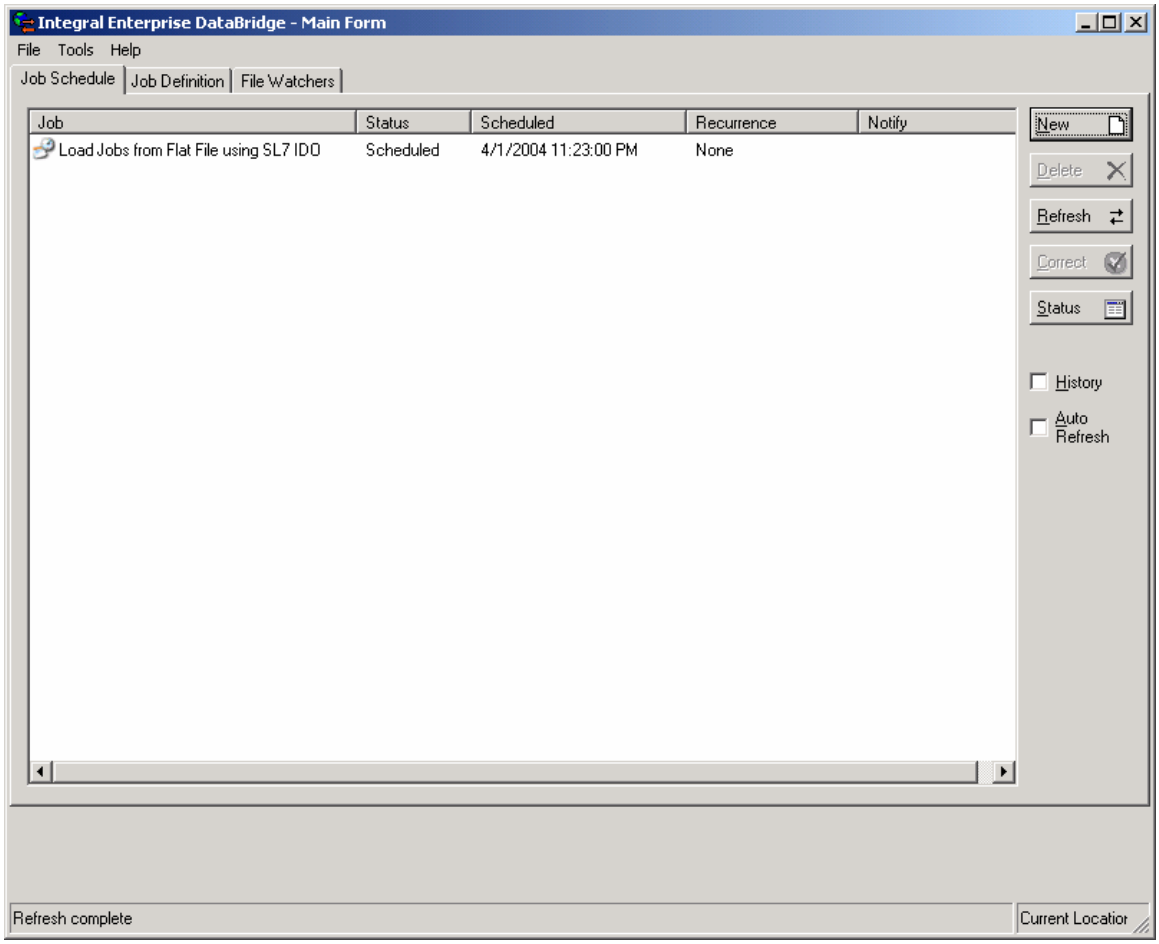
Notes:

- all scripts are standard VBScript
- the Initialize Script runs once when the job is initially started. Use this script to create any necessary objects (such as the SyteLine IDO's).
- the Data Script runs once for each row coming from the source data. This script is used to actually do the work of inserting data into SyteLine. Notice that the [<FieldName>] syntax is used to retrieve a literal value for the field being named.
- The Terminate Script is run once at the end of the entire job – use this script to destroy any objects used and perform any other necessary clean-up.

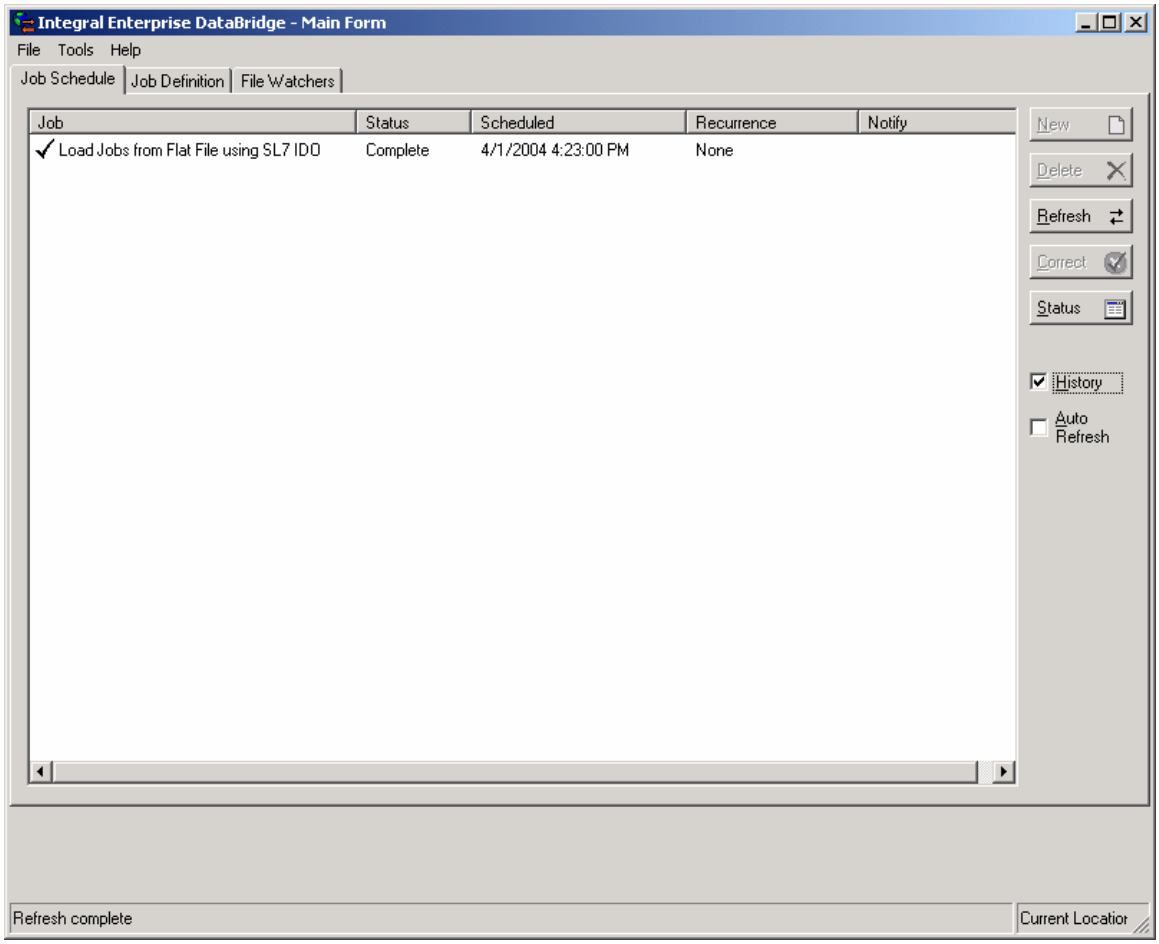
7) OK! Now that our Job is defined, we simply switch over to the "Job Schedule" tab and schedule the job to run. In this case, I want it to run immediately; but of course this operation could be scheduled to run off-hours, at certain scheduled recurrences, or even when the flat file simply is copied into the directory being "watched".



8) After we press OK, we simply wait for the Job to be run (it should begin in less than one minute).

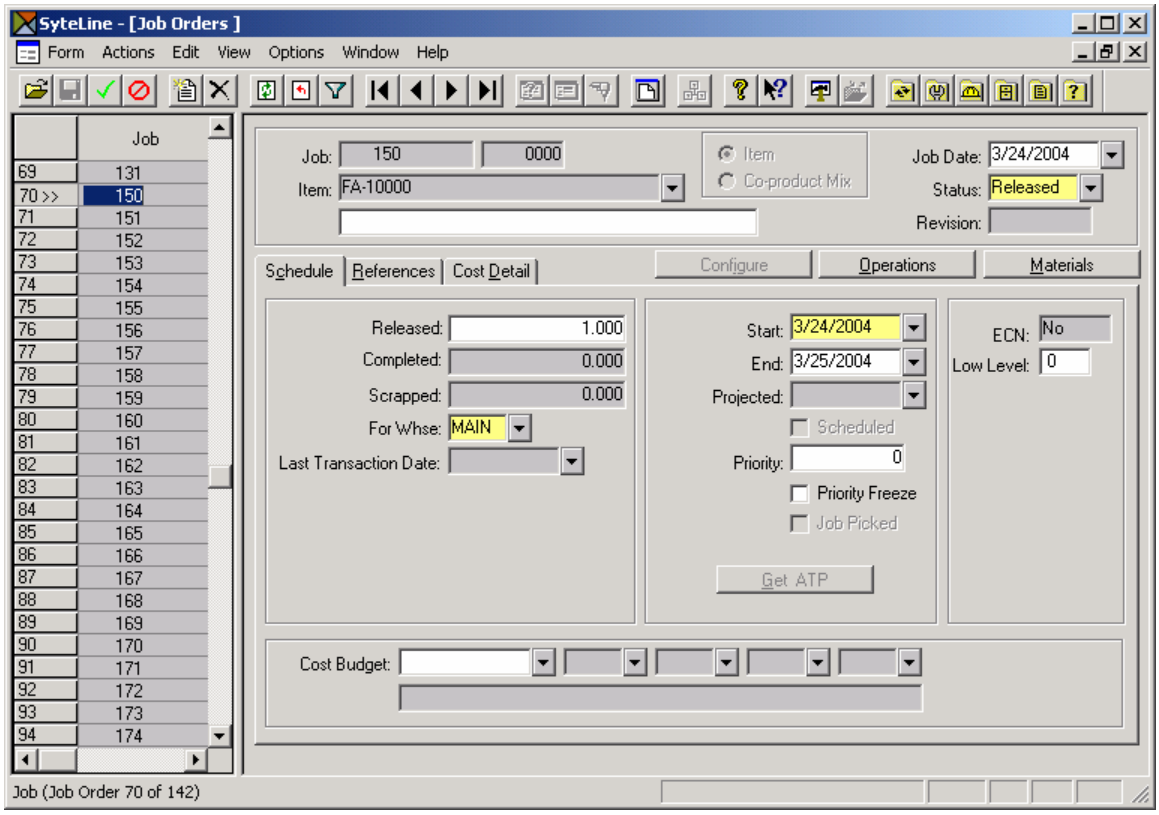


9) After the Job is run, it will disappear from the “Scheduled” view, and will instead appear in the “History” view. Check the “History” checkbox to view the history.



10) If there were any problems with loaded the data, the individual error rows can be updated and re-scheduled from the history view.

As you can see, the data loaded correctly into SyteLine:


SyteLine - [Job Orders]
Form Actions Edit View Options Window Help
Job
69 131
70 >> 150
71 151
72 152
73 153
74 154
75 155
76 156
77 157
78 158
79 159
80 160
81 161
82 162
83 163
84 164
85 165
86 166
87 167
88 168
89 169
90 170
91 171
92 172
93 173
94 174
Job: 150 0000
Item: FA-10000
Item
Co-product Mix
Job Date: 3/24/2004
Status: Released
Revision:
Schedule References Cost Detail
Configure Operations Materials
Released: 1.000
Completed: 0.000
Scrapped: 0.000
For Whse: MAIN
Last Transaction Date:
Start: 3/24/2004
End: 3/25/2004
Projected:
Scheduled
Priority: 0
Priority Freeze
Job Picked
Get ATP
ECN: No
Low Level: 0
Cost Budget:
Job (Job Order 70 of 142)