

Enterprise Output Manager

**UCSVmap
Guide**

UNISYS

© 2010 Unisys Corporation.
All rights reserved.

Release 1.4

Printed in USA

February 2010

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product or related information described herein is only furnished pursuant and subject to the terms and conditions of a duly executed agreement to purchase or lease equipment or to license software. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Notice to U.S. Government End Users: This is commercial computer software or hardware documentation developed at private expense. Use, reproduction, or disclosure by the Government is subject to the terms of Unisys standard commercial license for the products, and where applicable, the restricted/limited rights provisions of the contract data rights clauses.

Correspondence regarding this publication can be e-mailed to doc@unisys.com.

Unisys is a registered trademark of Unisys Corporation in the United States and other countries.

All other brands and products referenced in this document are acknowledged to be the trademarks or registered trademarks of their respective holders.

Contents

Section 1. Introduction.....	1
Section 2. Installation and Configuration.....	2
UCSVmap Custom Job Attribute.....	2
Parameterized Options.....	3
Program Operation Options.....	3
Creating the Form File Utility File	4
Required Output Manager Configuration.....	5
Error Handling.....	6
Example.....	7

Section 1

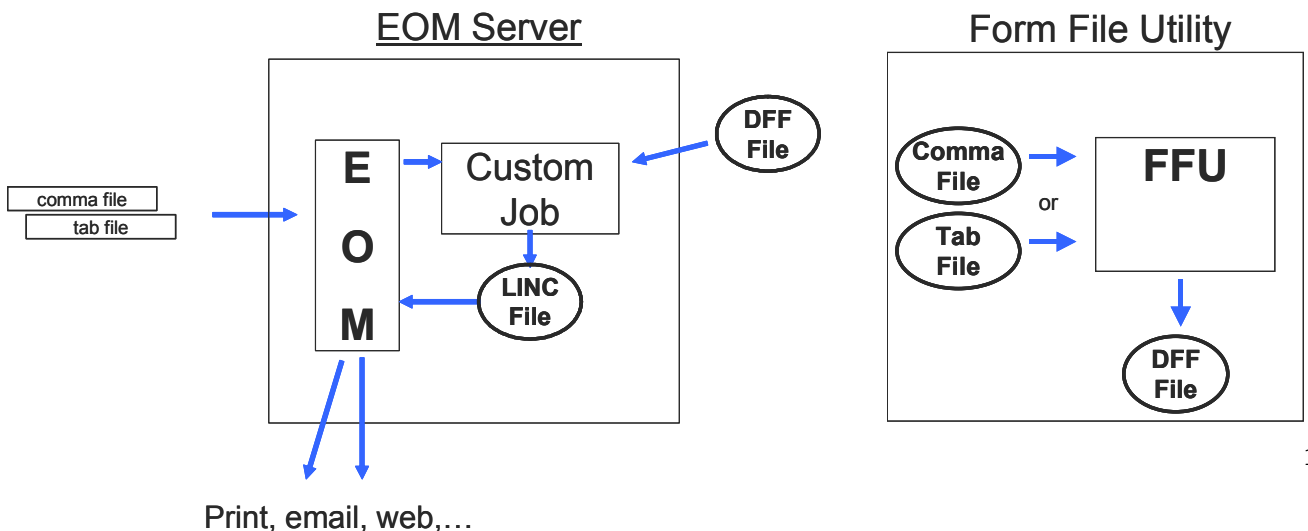
Introduction

The UCSVmap program, configured as an Output Manager Custom Job, is used to process comma or tab delimited records into Form File Utility composed pages. This solution provides a simple, visual way to place data if the original input file sent to Output Manager contains records of data where individual fields are separated by either a comma or a tab character. Use of this Custom Job does not require any change to the application producing the file.

UCSVmap can be seamlessly added to existing Output Manager Release 7.0 and later installations, requiring only the addition of Output Manager configurations to direct files to the program (File Mask), start the Custom Job (Custom Job Attribute), and then recognize & format the data stream returned from the UCSVmap program (File Mask, Print Attribute).

The following figures show the process when using the UCSVmap program. Output Manager receives tab or comma delimited files from various remote and local sources, including ClearPath hosts, UNIX systems, Linux systems, Windows applications, and file directories. Depending on the Output Manager configuration, some or all of these files are passed to UCSVmap (an Output Manager "Custom Job"), which Output Manager executes once for each file.

On each execution the UCSVmap program sends a processed input file in the "LINC" format back to Output Manager (using the Output Manager COM interface) which is then composed into the final output using the "LINC" Data Dependant Attribute (DDA) and the Form File Utility file (.DFF). The placement of data is controlled by the Form File Utility file, which is created by the user prior to execution of the UCSVmap Custom Job.



Section 2

Installation and Configuration

Installation is performed by manually copying the UCSVmap.exe program to an appropriate directory on the EOM PC. For example, the UCSVmap.exe file might be placed in a directory named c:\CustomJobs.

Configuring the UCSVmap Custom Job requires steps on the EOM PC. You must create one or more Output Manager File Masks that define which files sent to Output Manager create UCSVmap Custom Jobs, a Custom Job Attribute to execute the UCSVmap program, one or more Print Attributes defined with the "LINC" DDA to compose the output, and must import the "LINC" DDA prior to Custom Job execution.

Prior to executing the UCSVmap program, you must use the Form File Utility to create a .DFF file that contains the mapping of individual fields from the input comma or tab delimited file to the position on the output page. Each record of the input file is assumed to represent one "page" of information.

When the UCSVmap program executes, the values for fields from the input file are automatically placed on the Form File. Note that a .DFF file name is a required parameter for the UCSVmap program.

UCSVmap Custom Job Attribute

A Custom Job attribute defines what program to execute and what parameters are to be passed to that program. In addition to the name and location of the UCSVmap executable, the command line for UCSVmap has required parameters and optional parameters. The format of the command line is:

```
UCSVmap.exe I1=InFile D1=OutDir DFF=abc.dff (opts)
```

An example Custom Job Command Line is:

```
c:\CustomJobs\UCSVmap.exe I1="<PCFILENAME>" D1=f:\t1 dff=abc.dff  
HB=<BANNER> HR=<RUNID> HQ=<HOSTQUEUE> HU=<USERID> /L
```

In the example above, the input file "<PCFILENAME>" is the file received by Output Manager, which will be processed into output directory "f:\t1", using the Form File Utility file "abc.dff", and optionally passes a few values to be returned when UCSVmap submits the processed file back to Output Manager for composition.

Parameterized Options

<u>Option</u>	<u>Description</u>
I1=InputFile	is a file name pointing to the original input file
D1=OutputDir	is a directory where the processed file is created
DFF=abc.DFF	is a DFF file name used to position the input data fields
DA=MyDir	is the Directory Alias name for the DFF file (optional, if the DFF file is NOT in ../Output Manager/prntctrl directory)
HS=	Defines the EOM Mask Field value HOSTSYSTEM
HQ=	Defines the EOM Mask Field value HOSTQUEUE
HR=	Defines the EOM Mask Field value RUNID
HQL=	Defines the EOM Mask Field value QUALIFIER
HFL=	Defines the EOM Mask Field value FILENAME
HA=	Defines the EOM Mask Field value ACCOUNT
HB=	Defines the EOM Mask Field value BANNER
HP=	Defines the EOM Mask Field value PROJECT
HU=	Defines the EOM Mask Field value USERID
HOF=	is used for logging (input use is ORIGFILENAME)

Program Operation Options

<u>Option</u>	<u>Description</u>
/d	to delete the InputFile when done writing to output directory
/nw	for no wait -- disable single thread
/L	to log the Custom Job run in a text file
/LEL	to log errors to the Event Log

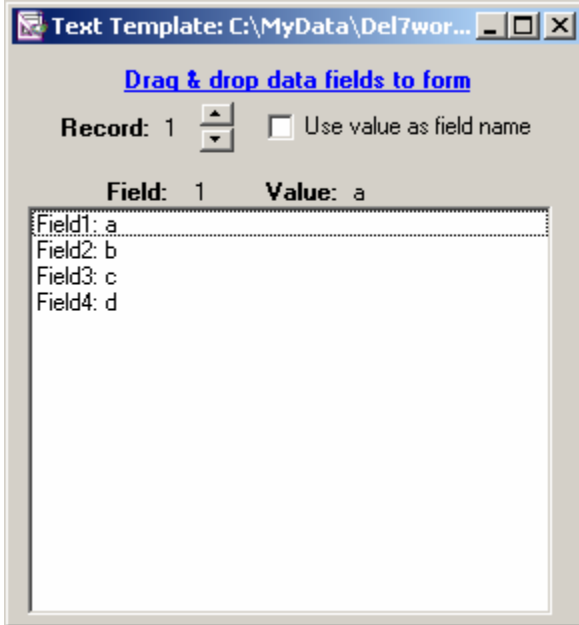
Creating the Form File Utility file

The Form File Utility allows you to graphically place fields on an electronic form and to supply the characteristics of those fields. The output from this utility is a .DFF file, which is composed of a Windows Metafile (WMF) or Windows Enhanced Metafile (EMF) and a collection of fields.

The Form File Utility enhances the Data Dependent Attribute (DDA) Designer feature by making it easier to use. The Form File Utility provides a way to specify the output location for data that does not require specific X and Y coordinates when composing the printed page.

When used with the UCSVmap Custom Job, the Form File Utility allows the user to place fields from a comma or tab delimited file onto an electronic form. The steps to create a Form File Utility file to process tab or comma delimited files are as follows:

- If not previously done, locate the Form File Utility initialization file (FFU.INI) and add the string "ShowCSVMenu=1" under the "[Flags]" section of the initialization file. Run the Form File Utility program and you will now see a "File / Open Text Template ..." menu selection.
- (optional) Use the menu selection "Edit / Load a Metafile" to add an electronic form to the Form File Utility file.
- Using the "File / Open Text Template ..." menu selection, open an example comma or tab delimited input file that has exactly the same data layout characteristics as those files to be processed in production.
- Select one field at a time from the Text Template dialog box and then drag & drop onto the Form File Utility form.
- You can then specify specific field properties as desired by selecting the individual fields and modifying the Property Inspector properties.
- Save the Form File Utility file (.DFF) in a directory of your choice or the Enterprise Output Manager /prntctrl directory.



The Text Template dialog box initially displays the first record data (in this case "a,", "b", "c, "d"). Simply select the particular field, then drag & drop onto the form. Note that if you do not recognize the field layout of the file by the first record data, you can use the "Record: _" scroll buttons to locate data within the file that you do recognize.

By default the name of the drag & dropped field will be Field x where " x " is the sequential field number of the comma or tab delimited record. When the "Use value as field name" checkbox is checked the drag & dropped field will be named the same as the record field data (for example it would be named "a" instead of

"Field1". In many comma and tab delimited files the first record of the file is a template with field names, so it is useful to have those names as the Form File Utility field names.

Required Output Manager Configuration

Aside from the File Mask that submits the input file to the UCSVmap Custom Job Attribute, other Output Manager configurations are required when using UCSVmap.

A Print Attribute may be required to compose the input data file stream prior to sending the input data to the UCSVmap program. A typical example is when there are Output Manager specific code sequences sent from the MCP or O.S. 2200 host that are to be removed prior to writing the output file. This Print Attribute is typically defined with a "Generic" or "Generic-NoFF" Printer Driver. If a Print Attribute is used, the pseudo printer "USER-PR" must be enabled.

Note that if a Print Attribute is defined on a Custom Job, Output Manager composes the data stream in a temporary file that is passed to the Custom Job. Typically it is the Custom Job's responsibility to dispose of this temporary file, which can be accomplished using the "/d" option. Please note that if you aren't using a Print Attribute but did use the "/d" option, then UCSVmap will dispose of the original input file, usually an undesirable action.

The File Group settings must be set "long enough" for the Custom Job to access the file. This is not an issue when using a Print Attribute because a temporary file is created. Without a Print Attribute, the original file name is typically passed to UCSVmap and unless the File Group is set to hold onto the file for a duration it is possible that Output Manager will remove the file before the UCSVmap program is done with it. As a general rule, when not using a Print Attribute, set the File Group to remove the file from the Output Manager database after 1 hour.

When the UCSVmap program executes, it creates a new file processed based on the input file and Form File Utility file (.DFF) parameter. This new file is given back to Output Manager via the COM interface MASK call. So, at least one File Mask should be created to "catch" the files returned from the UCSVmap program. The Host System parameter is always set to "UCSVmap" on files submitted to EOM by UCSVmap. The File Mask will typically be defined with a Print Job to print the processed file so a Print Attribute is required. That Print Attribute MUST be defined to use the "LINC" Data Dependant Attribute.

Error Handling

If the UCSVmap program fails or if the "/L" logging option is used, an entry is added to the UCSVmap log file, which can be found in the same directory as the UCSVmap program. The log file entry will describe the failure and include the entire syntax used to execute the Custom Job. There can be a log file for each day, where the filename looks like:

UCSVmapLog040210.txt

Where "04" is the year, "02" is the month, and "10" is the day this instance of UCSVmap executed.

The Output Manager Event Log will also contain entries describing the failure, Use the Log Viewing & Report tool to view these messages if the real-time Event Log has scrolled beyond the error messages.

If the Custom Job property "Disable job if application fails" is set to "Yes" (EOM 8.1 and above) and there is a failure, an Alert will be set with the exit code for the type of failure:

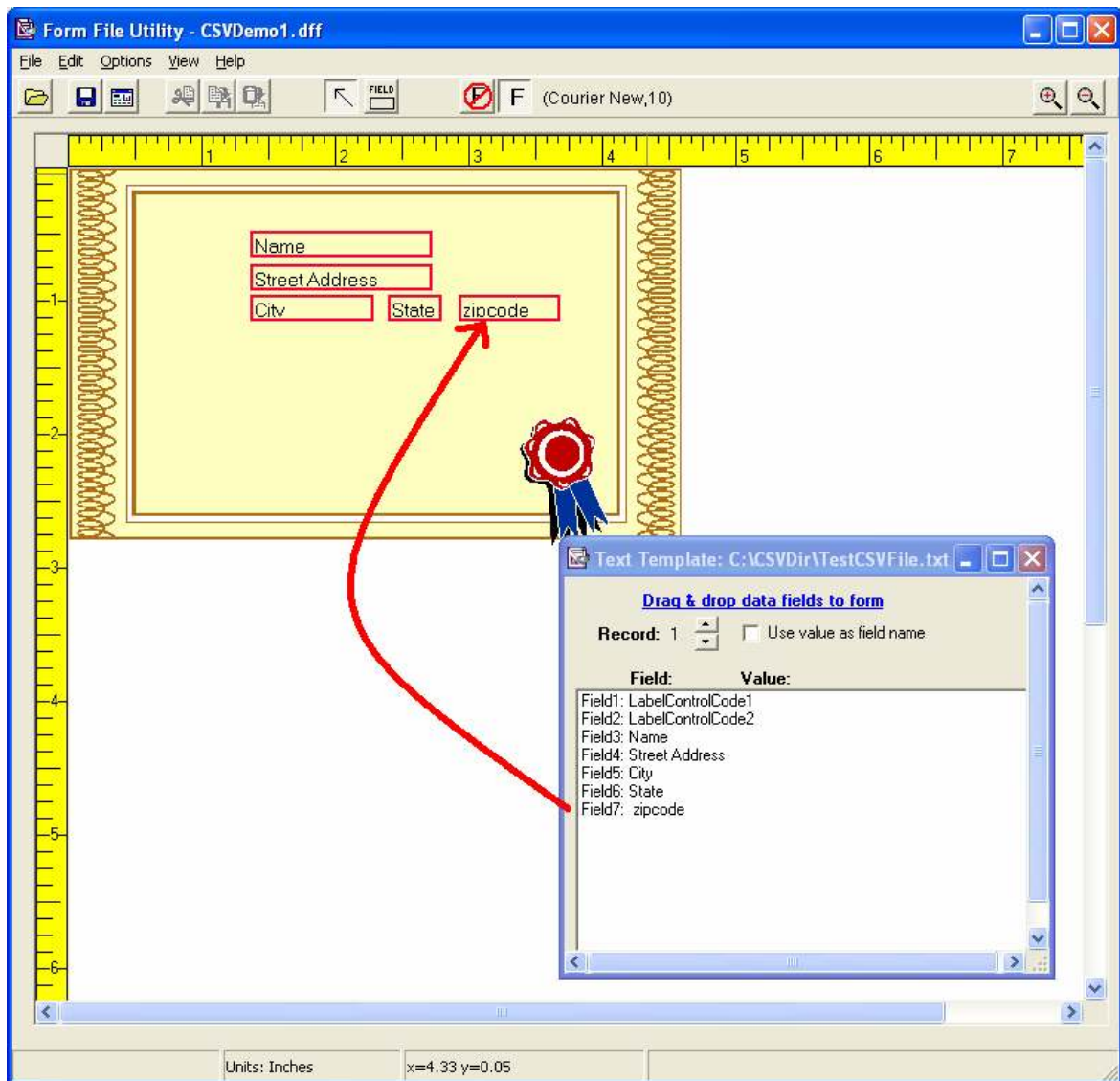
- 0 - No failure, program completed normally.
- 10 - Problem with the command line syntax used.
- 11 - Program started, but there was a processing issue.
- 20 - The program encountered an exception.

Example

The following is a simple example showing how to configure UCSVmap for the demonstration file. Suppose you have to process the following tab delimited file, where each record represents one page of printed output:

```
LabelControlCode1,LabelControlCode2,Name,,City,State, zipcode
#BXBGDD***5-DIGIT5 5041,#HAG0601 214408#SEP99 8401,HOMER HAGERSNASTY,2134 PITH WAY,ROSEVILLE,MN,55041-2345
#BXBGDH***5-DIGIT 43241,#PHO0504 214408#JAN98 8401,SALLY PHONEBONE,23 POLYNOMIAL BLVD,OPHIR,AL,43241-2345
#BXBBGER***3-DIGIT 117,#TRE0504 214408#JUN97 8401,ODNEY THRENIN,13452 KINETIC DRIVE,BOAZ,AL,11741-2345
#BXBBGHJ***5-DIGIT 67541,#HOL0504 214408#JAN98 8401,SCARY HOLIEN,789 STUCKGUM ROAD,RIDGECREST,CA,67541-2345
    .      .      .
```

There are six "fields" of information on each line, we use the Form File Utility to drag & drop those fields to a Form File Utility file:



The Custom Job command line would look something like:

```
I1="<PCFILENAME>" D1=c:\CSVDir DFF=c:\CSVDir\CSVDemo1.dff
```

There would be two File Masks, one to identify the original input file and one to identify the processed file returned from UCSVmap. In this case the File Mask used to identify the original input file creates a Custom Job without a print Attribute. The second File Mask creates a Print Job using a Print Attribute defined with the "LINC" DDA"