Tech Note 112

Recovering a Corrupted InTouch Application

All Tech Notes and KBCD documents and software are provided "as is" without warranty of any kind. See the **Terms of Use** for more information.

Topic#: 000720
Created: October 1998

This *Tech Note* outlines the most common reasons Wonderware® InTouch™ applications can become corrupted and provides a step-by-step procedure to rebuild and recover your application.

> **Note** You should always backup your InTouch application files. If a backup of your application is not available and the DBDUMP.EXE utility is unable to create a CSV file from your application database, then only a partial restoration of your application will be possible and your database must be recreated from scratch. In such cases, the windows and scripts can often still be recovered.

Identifying a Corrupted Application

Most InTouch application corruption occurs after an application's windows, tagname database, or scripts have been modified. You will know your application is corrupted primarily because a Wonderware Technical Support engineer has determined that this is the case. Here are some common indications of a corrupted application:

- Opening a window or running a script causes WindowMaker to terminate or shut down abnormally.
- When starting WindowViewer, you see an assertion error. Assertion errors indicate that there is a problem with the database entries in the animation links or scripts. Typically, assertion errors appear as FDRD errors. FDRD is the prefix used in the error message dialog box, such as **FdRdKey element #1**.
- WindowViewer starts and runs properly for a while and then suddenly, for no apparent reason, a GPF or an Assertion error occurs. Usually this happens when a corrupted script executes on the change of an event or a corrupted window is called to open.
- During the conversion of your application from one version of InTouch to another, the conversion halts with an error.

> **Note** If the conversion halts as described above, move the error message aside and take note on which window or what type of script the conversion program stopped. Most likely, that window or script is corrupted.

Three Rules for Avoiding Application Corruption

The primary cause of application corruption is copying or moving an application from one directory to another, such as copying an application from a development machine to a runtime machine. Here are three rules to keep in mind when copying or moving applications:

Note The following rules are most commonly violated when an application developer backs up an application. Be sure to keep all three of these rules in mind when backing up an application.

All of the files for an InTouch application must be considered as a single entity for copying purposes. No files should be individually separated from the whole.

1. Assume that you are an application developer and you receive a call from a customer. He asks you to make a small modification to an application that is currently running on his factory floor. It seems clear that the fastest way to remedy this situation is to modify your copy of his application in-house, and then send the modified application to the customer. You make a backup of the application first, and then add the changes. A thought occurs to you—why send the entire 10-MB application to the site when only one window changed? Just to be safe, you open Windows Explorer and look through the application directory for any files with the current date. It appears that only three files have changed, and one of them is a backup, so you assume that you only need to send the two remaining files. The customer receives your floppy disk with the two files and follows your instructions to copy them into the application directory. However, when the customer goes to runtime, WindowViewer causes a General Protection Fault or an Assertion Error.

2. The destination directory must be empty before copying an application. If you plan to copy an application, first rename the existing application directory and then create a new directory that has the original directory's name. Copy all of the application files into this new directory. If you have another backup of the application, and you are confident the new application is running properly, you may delete the original directory

3. Do not directly add, delete, or modify (through the MS-DOS command prompt, Windows Explorer, or File Manager) any files in the InTouch application directory, unless they are listed in *Tech Note* 85, titled "Files That Can Be Deleted From an InTouch Application," or you have been instructed to delete them by a Wonderware Technical Support engineer. The only supported methods of adding, deleting, or modifying application files are through WindowMaker.

Note The development computer and the runtime computer must have the same language configured under the system regional settings. This is because each language sorts special characters and their substitutes differently. For example, "AA" will be sorted last, instead of first, under the Norwegian regional settings. Also, if the same language is not configured under the regional settings for both computers, the index to the database will be wrong. (If this is the cause of your currupted application, you may be able to fix it by deleting the AVL and BIN files from your InTouch application directory.)

Procedure to Recover a Corrupted Application

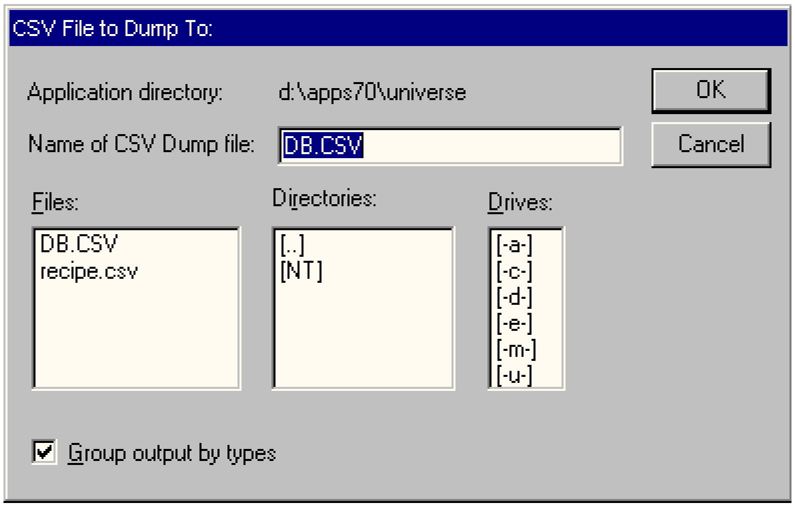
Wonderware Technical Support has found the following procedure to be the most effective way to recover a corrupted application and recommends that you follow these steps in order. Do not follow these steps out-of-order unless you are directed to do so by a Wonderware Technical Support engineer. Note that the term "legacy application" in this section refers to the directory that contains the corrupted application files, which should be considered as a single entity.

Step 1: Make a backup copy of your legacy application files

When you backup your legacy application files, be sure to follow the three rules listed above for avoiding a corrupted application.

Step 2: Create a CSV file of your tagname database using DBDump.exe

In InTouch version 7.0, DBDump.exe can be launched from the toolbar in the InTouch – Application Manager. In InTouch versions prior to 7.0, DBDump.exe is located in the InTouch installation directory and can be launched by selecting **Start/Programs/InTouch for Windows**. In these older versions of InTouch, DBDump will prompt you for the source application directory; however, in InTouch version 7.0, the source application directory will already be determined by the item that is selected in the main window of the Application Manager. Regardless of the version of InTouch you are using, be sure to verify that the source application directory points to your *legacy* application directory. The name of the CSV file you create is arbitrary as long as you remember the name of the file when you are prompted for it later. You should create the CSV file in the same directory as the legacy application. Here is the final DBDump.exe dialog box:



CSV File to Dump To:

Application directory: d:\apps70\universe

Name of CSV Dump file: DB.CSV

Files:
DB.CSV
recipe.csv

Directories:
[..]
[NT]

Drives:
[-a-]
[-c-]
[-d-]
[-e-]
[-m-]
[-u-]

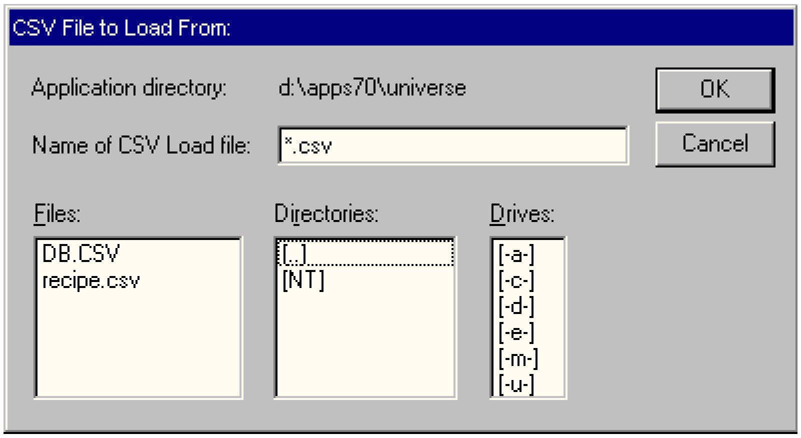
☑ Group output by types

Step 3: Create a new InTouch application

Open InTouch, select **New** from the **File** menu, and create a new InTouch application. Launch WindowMaker and then immediately exit WindowMaker. The purpose of this step is to create the application starter files. At this point, you have not yet created a database, scripts, or graphics in the new application.

Step 4: Load the CSV file using DBLoad.exe

Make sure you have exited WindowMaker and then load the CSV file of your tagname database using DBLoad.exe. In InTouch version 7.0, DBLoad.exe can be launched from the toolbar in the InTouch – Application Manager. In InTouch versions prior to 7.0, DBLoad.exe is located in the InTouch installation directory and can be launched by selecting **Start/Programs/InTouch for Windows**. In these older versions of InTouch, DBLoad will prompt you for the source application directory; however, in InTouch version 7.0, the source application directory will already be determined by the item that is selected in the main window of the Application Manager. Regardless of the version of InTouch you are using, be sure to verify that the source application directory points to your *new* application directory. Here is the final DBLoad.exe dialog box:



CSV File to Load From:

Application directory: d:\apps70\universe

Name of CSV Load file: *.csv

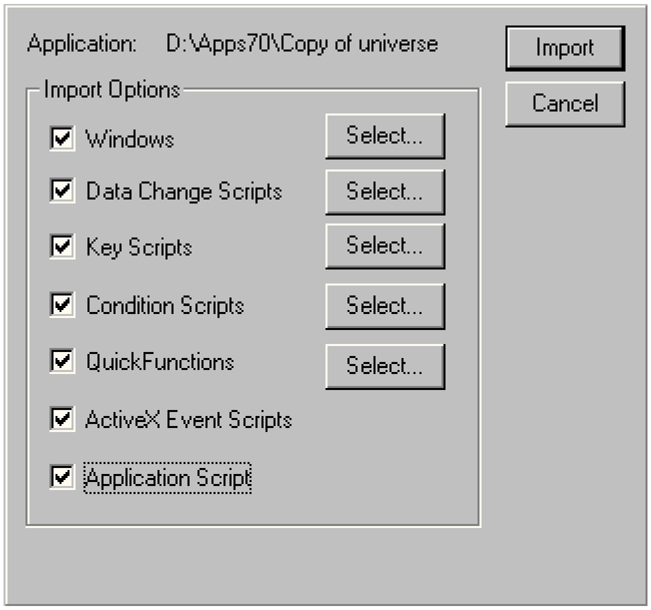
Note Often it is during this stage that problems are encountered with tags that will not load. If an ASK dialog box appears, there is probably a problem with the alarm groups. Click **Abort load** and then acknowledge the *Load Failed* dialog box. Open the WWLogger and observe the DBLoad failure line (generally at the bottom), writing down the line number where the failure occurred. Open the DB.CSV file in a worksheet editor such as Microsoft Excel. Find the row number that matches the line number you noted from the WWLogger and then fix the problem in this row as reported by the WWLogger.

Step 5: Verify the database in WindowMaker

Launch WindowMaker and verify that you have a new application containing your legacy database. For example, open the tagname dictionary and select a few tags that you know should be in the database and verify that you can see them. At this point your application should not contain any graphics or scripts.

Step 6: Import all windows and scripts from the legacy application

Anything you want, but do not import, you will have to create from scratch. To import the windows and scripts, select **Import** from the **File** menu. Then, select the legacy application directory and click **OK**. You will see the following dialog box (this dialog box may appear differently depending on your version of InTouch):



Application: D:\Apps70\Copy of universe

Import Options

☑ Windows

☑ Data Change Scripts

☑ Key Scripts

☑ Condition Scripts

☑ QuickFunctions

☑ ActiveX Event Scripts

☑ Application Script

Make sure **Application** is pointing to your legacy application path. Next, select all of the boxes, and for the Windows, Data Change Scripts, Key Scripts, Condition Scripts, and Quick Functions, you also need to click the **Select** button and select all of the items. For example, selecting the box labeled **Windows** will *not* select any windows for importation. Click **Import** when you have selected everything you wish to import. It may take a while to import everything, depending on the number of items that are being imported.

Warning InTouch 7.0 uses a 60K tagname license, which is larger than the tagname license for previous versions of InTouch. As a result, this has reduced the placeholder database to a maximum of 4,095 tagnames. If you are using InTouch 7.0 and your legacy application has more than 4,095 tagnames, pay special attention to the following information. Larger legacy applications will need to be divided into smaller logical pieces. Each piece must have fewer than 4,096 tagnames. **Example 1**: You might estimate that all of your scripts can be imported at once, with less than 4,096 tags. After importing this piece you must convert all the placeholders. Doing so will free up space in the placeholder database for the next imported piece. Placeholder conversion is explained in Step 7 of this *Tech Note*. **Example 2**: Say you have 20,000 tagnames in your application with 200 windows. Perhaps you can safely import 25 windows at a time. Convert the placeholders in these windows, and import the next 25 windows. Since WindowMaker will not track this information for you, remember to keep an accurate record of which windows and scripts you have already imported. On a final note, if you have a single window or script which contains more than 4,095 tagnames, contact WonderwareTechnical Support for assistance.

Step 7: Convert all placeholders

When you imported the windows and scripts, placeholders were added to the tagnames. The placeholders are the first three characters of the tagname: a question mark, followed by an alphabetical character that represents the data type, and then a colon. For example ?d:, ?i:, ?r:, and ?m: respectively stand for discrete, integer, real, and message. Follow these steps to remove the placeholders so the tagnames can be linked to the database:

1. Open one of the imported windows and select all of the items on the window by selecting **Select All** from the **Edit** menu.

2. Select **Substitute Tags** from the **Special** menu and then remove the placeholders by clicking the **Convert** button. (If you are using InTouch 7.0, you will have the option to convert local or remote tags. Since we are linking the placeholders to the local database, click the **Local** button.) Then click **OK**.
3. Select **Window/Properties/Scripts** or **Special/Scripts/Window Scripts** to open the window script editor. Click **Convert** (in InTouch 7.0, click the **Local** button next) and then click **Save** to automatically convert all placeholders for all three window script types: On Show, While Showing, and On Hide.
4. Save the window and close it.
5. Now repeat the steps above for each window in the application that you imported.
6. Next, select **Special/Scripts/Application Scripts** to view the Application scripts. Remove the placeholders by clicking **Convert** (in InTouch 7.0, click the **Local** button next), and then click **Save**.

Repeat this procedure for each of your Data Change scripts, Condition scripts, Key scripts, and for InTouch version 7.0, your QuickFunctions.

Step 8: Copy InTouch.ini from your legacy application and overwrite the InTouch.ini in the new application.

This will ensure that all of your configuration options have been preserved. Note that this will not update the home window configurations however. To configure the new application, select **Special/Configure/WindowViewer/Home Windows**. Most of your configuration options are contained in the InTouch.ini file.

Note that you can also copy the almgrp.ini and histprv.ini files, which contain information on how the alarm and history providers are configured.

Step 9: Before running WindowViewer, backup your application.

When you backup your application files, be sure to follow the three rules at the beginning of this *Tech Note* for avoiding a corrupted application.

Step 10: Run WindowViewer to compile your application

After you have compiled your application, spend some time to verify that your application works as it did before. Navigate all the various screens and, if possible, simulate some scripting events. Wonderware Technical Support recommends that you keep a permanent backup copy of your legacy application. This is best done with backup media. However, you should keep a backup on the local hard disk for a little more than one month, then you can clear up the hard disk space later if necessary.

M. McCulty

The *Tech Note* is published occasionally by Wonderware Technical Support. Editors: Sabrina Abercromby and Kelly Hauser; Publisher: Wonderware Corporation, 100 Technology Drive, Irvine CA 92618. There is also technical information on our software products at **Wonderware Technical Support**, our WonderFax fax-on-demand system at (949) 450-5050, and the Comprehensive Support Knowledge Base.

For technical support questions, send an e-mail to **support@wonderware.com**.

back to top