

---

# Verifying an Intelligent Search Upgrade

This document provides instructions for using the InQuira UpgradeTool utility to verify that an existing Intelligent Search installation contains all of the required files following an upgrade from 8.1.2.x to 8.1.3.3 and explains how to repair installations if they fail verification. Errors are fixed by copying the correct files from a fresh install of the target version and by removing files that should have been deleted during upgrade.

Download the Search Upgrade Cleanup Utility utility (UpgradeTool.zip) from the InQuira web site. The Search Upgrade Cleanup Utility is available to download from

<http://supportfiles.inquira.com/support/Utilites/>

Unzip the file UpgradeTool.zip on the host of the upgraded search installation to be verified.

## Verify an Intelligent Search Upgrade

To verify an Intelligent Search upgrade, run the Verify Install script:

**Note:** On Linux systems, add execute permissions to the tool scripts:

- cd <UpgradeTool\_dir>/bin
- chmod 775 \*.sh

- 1 Copy the <UpgradeTool\_dir>\SearchUpgradeUtil.jar to the {Upgraded\_Installation\_root\_dir}/lib folder.
- 2 Copy the <UpgradeTool\_dir>\bin\verifyInstall script to any instance directory on each host that has a Search installation.
- 3 Create an ICE Window for that instance.
- 4 Run the verifyInstall script, with two arguments:
  - the target version
  - the repository folder path

This script writes the results to the console and to an audit file in the repository folder:

```
<repo_dir>/<VERSION>_<PLATFORM>_audit.txt
```

For example, on Linux 32 bit OS, run this command in ICE to verify an upgraded 8.1.3.3 installation, where the upgrade utility tool folder is at /UpgradeTool:

```
./verifyInstall.sh 8.1.3.3 /UpgradeTool/repo
```

This generates an audit file 8.1.3.3\_LINUX\_32\_audit.txt in the repository folder.

**Important!** The verifyInstall script writes its findings to the console and audit file, but makes no changes in the upgraded installation.

---

## Review the Audit File

Review the generated audit file from the previous section to see if the utility found any errors in the upgraded installation.

The audit file contains one line for the status of each file checked in the upgraded installation. The following status messages appear in the audit file:

- **INFO: MATCHED <file>**  
The file in the upgraded installation matched the target installation.
- **INFO: IGNORED <file>**  
The file in the upgraded installation was not compared to the target installation because it is customized during the installation process.
- **ERROR: FILE FOUND AND SHOULD BE DELETED <file>**  
The file in the upgraded installation should have been deleted.
- **ERROR: FILE NOT FOUND <file>**  
The required file does not exist in the upgraded installation.
- **ERROR: MISMATCHED <file>**  
The file content in the upgraded installation does not match the same file in the target installation.

## Repair an Upgraded Installation

If there are any errors found in Verify an Intelligent Search Upgrade step 4, these can be fixed by generating a fix it script that copies the correct (target) version of the files, from a new target installation, to the upgraded installation. Files replaced by the fix it script are copied to a rollback folder. A revert it script, which is generated with the fix it script, can be used to rollback the changes made by the fix it script, should this be desired.

To fix the upgraded installation using a fix it script, download and run a full install of the target version of Search in a temporary location.

In the Upgraded Installation:

- 1 Copy the `<UpgradeTool_dir>\SearchUpgradeUtil.jar` to the `<Upgraded_Installation_root_dir>/lib` folder.
- 2 Copy the `<UpgradeTool_dir>\bin\mkFixItScripts` script to any instance directory on each host that has a Search installation.
- 3 Create an ICE Window for that instance.
- 4 Run the `mkFixItScripts` script, with the following four arguments:
  - the target release version
  - the repository folder path
  - the path of the target version of Intelligent Search that you installed above
  - the path of a folder to which the original files will be backed up (for rollback later, if necessary).

For example, on Linux 32 bit OS, run this command in ICE for an upgraded installation, where the upgrade utility tool folder is at `/UpgradeTool`, and the roll back folder is at `/8.1.2.5_Rollback`:

```
./mkFixItScripts.sh 8.1.3.3 /UpgradeTool/repo /InQuira_8.1.3.3 /  
8.1.2.5_Rollback
```

This generates the `fixIt.{bat,sh}`, `revertIt.{bat,sh}` scripts, and the `<VERSION>_<PLATFORM>_audit.txt` in the repository folder.

The generated `fixIt.{bat,sh}` script completes the following actions:

- a** For each ERROR: MISMATCHED file in the audit log, the script copies the upgraded installation file to the rollback folder; then copies the target version of the file to the upgraded installation.
- b** For each ERROR: FILE NOT FOUND file in the audit log, the script copies the target version of the file to the upgraded installation.
- c** For each ERROR: FILE FOUND AND SHOULD BE DELETED file in the audit log, the script moves the upgraded installation file to the rollback folder.

The generated `revertIt.{bat,sh}` script restores the files in the upgraded installation with the original files, from the rollback folder.

**Note:** The `revertIt` script can only be run after running the `fixIt` script.

On Linux systems, add `execute` permissions to the generated scripts before they can be executed:

- `cd <UpgradeTool_dir>/repo`
- `chmod 775 *.sh`

**Important!** Before running the `fixIt.{bat,sh}` script, please review the contents of the script file to ensure that every recommended file should be copied or removed and that there are no custom files affected.

- 5** After running `fixIt.{bat,sh}`, it is recommended to create an ICE Window for each instance again, and run `buildWebapp` and `deployApp` if prompted.

## Apply the autoimport.xml

The final step is to use the proper `autoimport.xml` file to apply additions to the Intelligent Search configuration (`#.xml`). These would be the additions that have been made to the `#.xml` between the version being updated and the target version. It may be necessary to apply the `autoimport.xml` from the unzipped utility package.

To check which `autoimport.xml` should be used for the particular Intelligent Search upgrade path see the table at the end of this guide. The table defines the upgrade path, as the Y axis for `upgradeFromRelease`, and X axis for the `upgradeToRelease`, the value is the version folder name to look for the proper `autoimport.xml`.

Note after copying the target version of `autoimport.xml`, from the `UpgradeTool` folder, to the instance, you need to stop the instance, quit ICE, and enter ICE again for the `autoimport.xml` to be applied to the `#.xml`.

Example 1: An upgrade from 8.1.2.0 to 8.1.2.5 has the value of 8.1.2.4.1. So the proper `autoimport.xml` should exist in `<UpgradeTool_dir>\autpimport\8.1.2.4.1`.

Example 2: An upgrade from 8.1.2.3 to 8.1.3.3 has the value of 8.1.3.x. So the proper `autoimport.xml` should exist in `<UpgradeTool_dir>\autpimport\8.1.3.x`.

Example 3: An upgrade from 8.1.3.3 to 8.2.0 has the value of 8.2.0. So the proper `autoimport.xml` should exist in `<UpgradeTool_dir>\autpimport\8.2.0`.

---

## Appendix

The VerifyInstall.{bat,sh} and mkFixItScripts.{bat,sh} scripts use a set of four files:

- <VERSION>\_<PLATFORM>\_valid\_files.txt:  
List of required files for a specific platform + version, not including any site-specific files.
- <VERSION>\_<PLATFORM>\_file\_checksums.txt  
Copy of <VERSION>\_<PLATFORM>\_valid\_files.txt with each file path followed by a checksum for the file.
- <VERSION>\_<PLATFORM>\_deleted\_files.txt  
List of files that should not exist in this platform + version after upgrade.  
(e.g., When sharedinfra.jar was replaced with inquiria-shared-infra-1.1.jar)
- <VERSION>\_<PLATFORM>\_ignored\_files.txt  
List of file names that should not be compared for this platform + version.

These four files are delivered as part of the UpgradeTool under the repo folder. For 8.1.3.3, these files are available for the following platforms (OS):

- Windows 32-bit
- Windows 64-bit
- Linux 32
- Linux RedHat 4.6 64 bit

For example, To verify an upgrade to Search 8.1.3.3 on Windows 64 platform, the following four files are needed:

- 8.1.3.3\_WIN\_64\_valid\_files.txt
- 8.1.3.3\_WIN\_64\_file\_checksums.txt
- 8.1.3.3\_WIN\_64\_deleted\_files.txt
- 8.1.3.3\_WIN\_64\_ignored\_files.txt

After unzipping the 8.1.3.3 UpgradeTool.zip file, these files can be found in <UpgradeTool\_dir>\repo.

from/to	8.1.2.1	8.1.2.2	8.1.2.3	8.1.2.4	8.1.2.4.1	8.1.2.5	8.1.2.6	8.1.2.7	8.1.3.0	8.1.3.1	8.1.3.2	8.1.3.3	8.2.0.0
8.1.2.0	None	None	None	None	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.2.1	N/A	None	None	None	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.2.2	N/A	N/A	None	None	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.2.3	N/A	N/A	N/A	None	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.2.4	N/A	N/A	N/A	N/A	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.2.4.1	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.2.4.1	N/A	N/A	N/A	N/A	N/A	None	None	None	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.2.5	N/A	N/A	N/A	N/A	N/A	N/A	None	None	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.2.6	N/A	N/A	N/A	N/A	N/A	N/A	N/A	None	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.2.7	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8.1.3.x	8.1.3.x	8.1.3.x	8.1.3.x	8.2.0
8.1.3.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	None	None	None	8.2.0
8.1.3.1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	None	None	8.2.0
8.1.3.2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	None	8.2.0
8.1.3.3	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8.2.0