

AucConverter - Designdraft

Motivation:

There have been many different versions of Auctioneer and with almost each new version, more or less changes were made to its dataformat.

Writing a converter from version 3.0 to 3.2 was really a match and disclosed many problems of a missing convertingsystem.

That's when the idea of designing a specific converter, which would keep all the trouble out of future versions of auctioneer came up.

Requirements:

The to be designed system has to fit the following requirements:

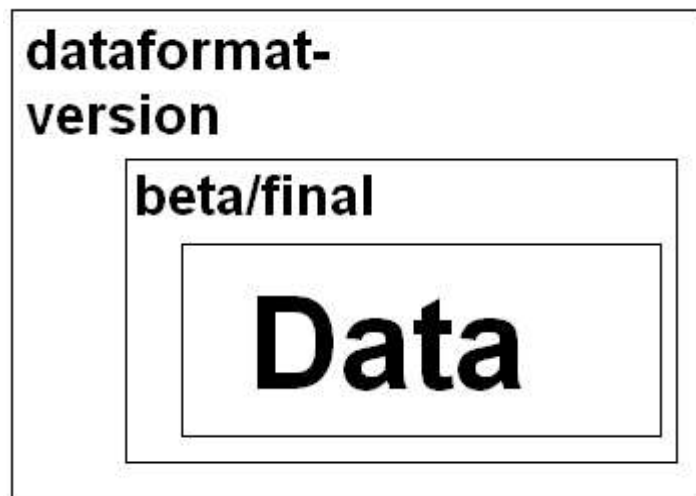
- allow upgrading from any older dataformat to the current format
- fixing bugs in converterfunctions with or without the lost of old data
- testing alpha-/betaversions must not affect the integrity of using a final release-version
- having little impact to the overhaul performance of auctioneer

Downgradingcapability is expicitly excluded from the requirementlists because:

- this reluts in much more work for the programmers
- usually users don't downgrade... They prefer to use the latest version available

Conception:

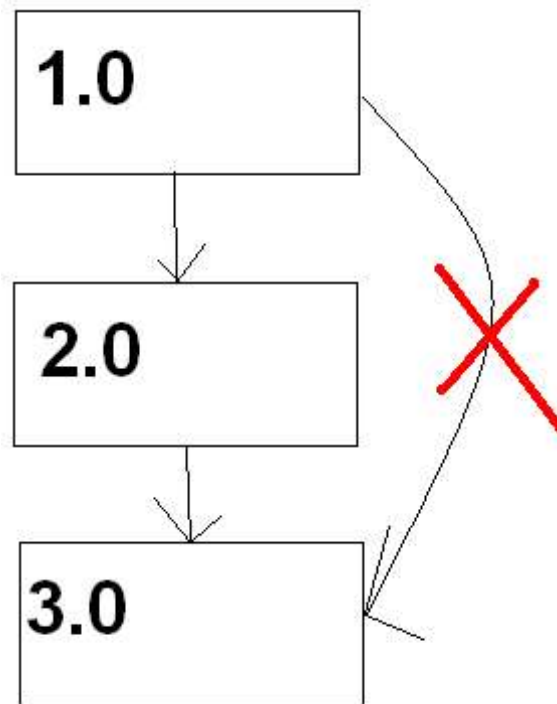
First, auctioneer's data will be encapsulated. Instead of storing only the data, also the dataformatversion and if the data is used by a beta or a final release is being stored now.



This way the data can be easily classified, instead of having to identify the dataformatversion by

analysing the data itself.

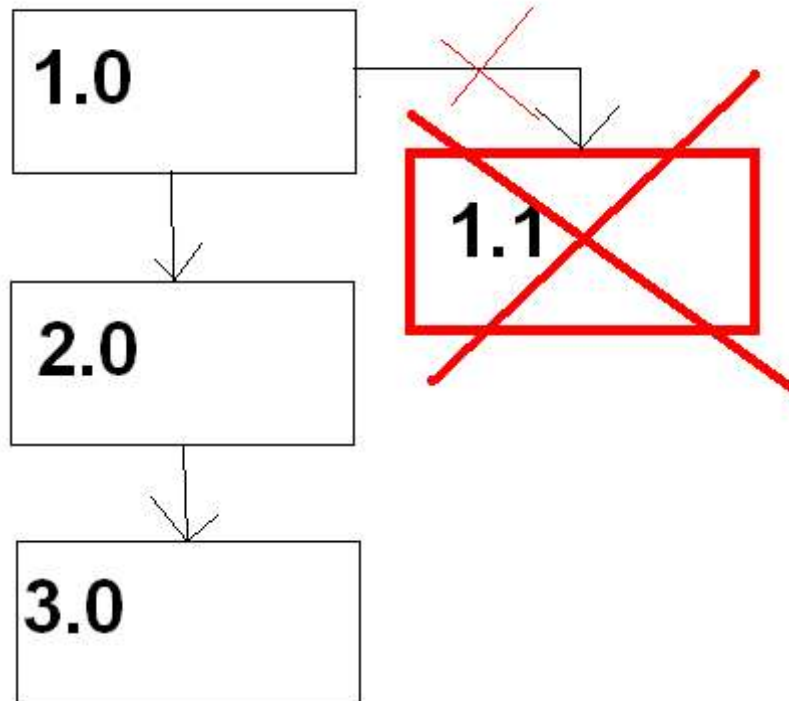
Second, converting older dataformats will be made modular.



Instead of scripting functions which convert the dataformat 1.0 to 3.0, the 1.0 dataformat would be converted to version 2.0 first and then into 3.0.

This results in less work for the programmers and less bugs (since only one function instead of many (for each old dataformat one) has to be scripted). The only con is, that this takes longer than a direct converting-function would. But since this procedure is only executed once, it's a minor issue.

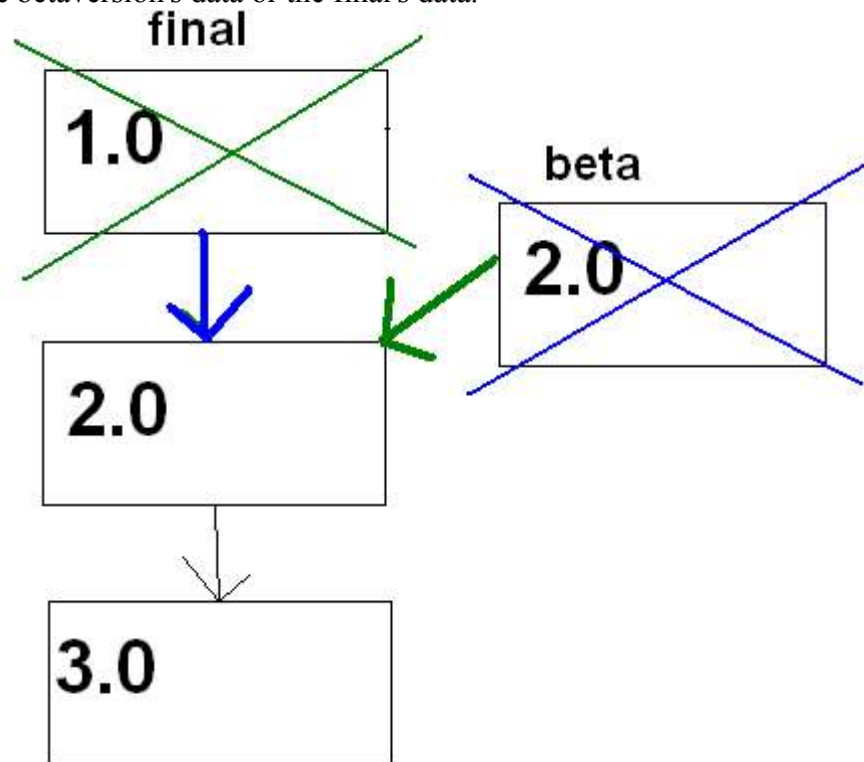
During (beta-)versions it might happen, that data is corrupted and can't be restored. One solution is, to keep backups of each dataformat (or at least a specific number of backups). As auctioneer data uses up to several MB this would mean to increase WoW's memoryusage by this amount. That's why there won't be an automatic backupfunction atm. When WoW offers a way to store data on the user's harddrive, this feature might be added. So, backing up the data is the user's task.



The converter allows to mark specific dataformats as corrupt. This data will not be converted, completely wiped out and the user has to start from scratch on.

Version 1.1 is blacklisted... When the converter find's a old 1.1 dataformat, it will be completely removed. The old converterfunction which converted version 1.0 to 1.1 no longer exists.

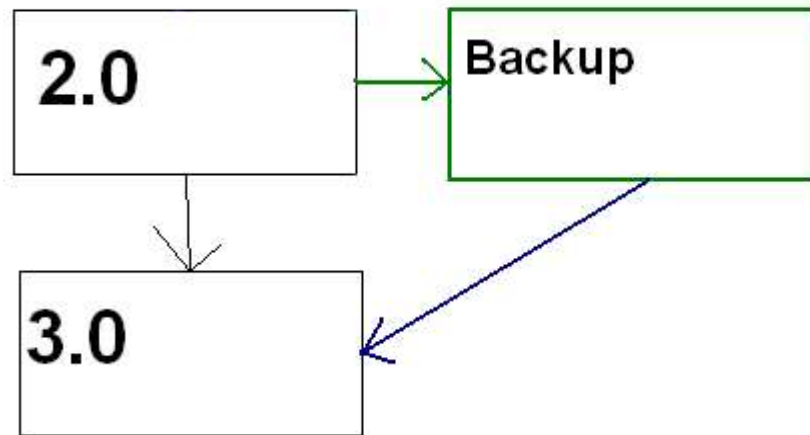
When converting to a new final dataformat, the converter lets the user decide which wether he wants to keep the betaversion's data or the final's data.



If the user prefers to keep the final data, the data is converted into the new format and the old betadata will be removed. Otherwise the old data is being wiped out and the old betadata becomes the new final data.

Converting the final data to the new dataformat and then merging the beta and final data might be possible, but would mean in a lot of extra work for each programmer. It's not only the merge-function which has to be added. Additional data must be saved for all auctions to find out duplicates later on. Since it is intended that people use final releases, only, this would be a waste of time.

Since some conversion procedures might take a long time (up to several minutes) or it is just not possible to convert part of the data atm (some data might only be accessible in specific areas, like when browsing the AH or talking to a vendor), data which can't be converted at the moment, will be backed up. The user's given a separate function to convert the data, when he wants to and special conversion functions will automatically convert backedup data, as soon as it's possible.



Implementation: