TRANS · núm. 15 · 2011
DOSSIER · 39-51

Enabling the Localization of Large Role-Playing Games

Chris Christou
Lead Localization Tools Programmer, BioWare

Jenny McKearney
Localization Producer, BioWare

Ryan Warden
Localization Project Manager, BioWare

In order to achieve total immersion in the game world, increasing therefore player' satisfaction, localization should ideally aim at creating complete suspension of disbelief. However, time constraints and constant design and script changes mean that localisation is sometimes forced to trade quality in favor of speed, because missing release dates can mean multimillion losses. This article explains the strategies BioWare has developed internally to counteract the problems provoked by long-established game development practices with the ultimate goal of supporting quality localization from the start, and so guaranteeing players' suspension of disbelief whatever the language version they play.

KEY WORDS: multiplayer, localization, role-playing, video game, online, MMO, RPG

Facilitando la localización de videojuegos de rol masivos

Con el objeto de lograr la inmersión absoluta en el mundo virtual del videojuego, aumentando así la satisfacción del jugador, la localización debe conseguir el ideal de la total suspensión de la incredulidad. Sin embargo, los cortos plazos así como cambios constantes de diseño y guión a veces obligan a que la localización tenga que cambiar calidad por velocidad, porque el cambio de las fechas de lanzamiento suele provocar pérdidas multimillonarias. Este artículo explica las estrategias que BioWare ha desarrollado internamente para contrarrestar los problemas provocados por las prácticas tradicionales en la industria del videojuego. El objetivo es facilitar un proceso de localización de alta calidad desde el principio del desarrollo, de modo que se garantice la suspensión de la incredulidad de los jugadores independientemente de la lengua en la que estén jugando.

PALABRAS CLAVE: *multijugador, localización, juego de rol, videojuego, en línea, MMO, RPG*

INTRODUCTION

It could be said that, in general terms, localization has been a part of human interaction since man first stepped out of the cave and attempted to communicate his experiences to people from other tribes. While that first attempt would have been decidedly crude, modern-day localization has the benefit of thousands of years of process-refinement. The fundamental quandary of localization hasn't changed though: how can one effectively communicate the myriad nuances of an experience to another party — especially to a party versed in, and indeed, expecting, a specific cultural manner?

This question becomes even more complicated when associated with interactive entertainment experiences such as video games. The mark of perfect localization would see a player considering a video game to have been created in his culture, for his culture. That is, the ideal localization would engender a complete suspension of disbelief.

Of course, the ideal is often unattainable. Time constraints reduce diligence in favor of speed. Budget constraints reduce creativity in favor of neutralized blandness. Scope constraints, on the other hand, often don't exist.

The presence of any one of these constraints severely undermines quality. In the development cycle of a given video game, there is a confluence of these (and more) constraints. To overcome these obstacles, BioWare has developed certain processes with the ultimate goal of enabling high-quality localization and maintaining the player's suspension of disbelief whatever the version they are playing.

BioWare

BioWare is a division of Electronic Arts that specializes in the creation of role-playing games (RPGs). Over the course of fifteen years BioWare has developed a number of critically acclaimed games: *Baldur's Gate*, *MDK2*, *Baldur's Gate 2*, *Neverwinter Nights* (and its myriad expansion packs), *Star Wars: Knights of the Old Republic*, *Jade Empire*, *Mass Effect*, *Sonic Chronicles: The Dark Brotherhood*, *Mass Effect: Galaxy*, *Dragon Age: Origins, and Mass Effect* 2. We're currently also developing a massively multiplayer online (MMO) RPG based in the Star Wars universe, entitled *Star Wars: the Old Republic*.

Role-Playing Games

All the games listed above are RPGs — games that, in addition to other gameplay elements, emphasize plot and character interaction through interactive dialogs, as well as character customization and progression. The final product is intended to present a fully-realized and compelling fantasy for the player. RPGs can, and should, be sprawling, epic universes. It is for this reason that developing RPGs is difficult.

The amount of content in an RPG can be staggering: *Dragon Age: Origins* contained well over 1,000,000 words and 56,000 lines of recorded voice-over audio (VO) in total. If one considers that an average novel contains approximately 75,000 words, *Dragon Age: Origins* was comparable in size to more than thirteen novels.

This paper will focus on the localization of the *Mass Effect* franchise, as the authors worked most closely on it. *Mass Effect* had approximately 300,000 words and approximately 25,000 lines of VO. *Mass Effect* 2 expanded on this scope, with 444,000 words and 30,000 lines of VO.

Additionally, each of the statistics quoted above pertains to the English version only. The localized versions of *Mass Effect* 2 added up to approximately 2.7 million words, and approximately 140,000 lines of VO (across four recorded languages: French, Italian, German and Polish). Over 300 days were spent in recording studios, utilizing the talent of more than 350 different voice actors in three different time zones.

LOCALIZATION

The scale of this sort of localization can be a challenge, especially given that our schedule dictates that we cannot wait until the game content is final before localizing it. BioWare games must be localized while content development is ongoing. If managing this sort of scope is difficult under the constraints of one language, it is especially so for multiple languages.

The essential localization workflow remains the same, despite the scope: export text for translation, translate, import text, print scripts (as necessary), record VO (as necessary) and repeat. The basic principles are nothing new, but when the demands of the game industry and the global marketplace are pushed onto traditional practice, many questions need to be raised: How can we refine the processes in this workflow? Can we automate some of them? Can we apply them to subsets of the game content? Can we re-apply them as necessary? Can we eliminate re-work where possible, or speed up the re-work where unavoidable? At BioWare we answer with a resounding «Yes».

Preparation

Usually in the early stages of game production, not enough information is available to allow us to make firm localization plans. There is no detailed scope information yet — often the only information available is «the last game plus or minus 20%.» This makes scheduling difficult, but there is still other project management work that can be done. One of our first steps is to put together as complete a localization kit as possible. The key idea is simple enough; translators need information to create high-quality translations, so we endeavor to provide as much information as possible. This has an added side benefit of eliminating uncertainty before it occurs.

Pronunciation Guide

We are lucky at BioWare in that the localization team works closely with the English (EN) VO team; reference material created for the EN team is often useful for the localization team (and vice-versa). This reference material typically includes a recorded pronunciation guide for the more esoteric terms in the intellectual property (IP). During the production of *Mass Effect* 2, for example, our senior editor worked with the EN VO team to record the pronunciation of various IP-specific terms, long in advance of the start of EN VO recording.

The localization team was able to re-use this work for the localized VO recordings, in an attempt to preserve consistency in the IP. There is no risk of stumbling over strange pronunciation in the recording studio; with such a pronunciation guide available, the studio should be able to focus on obtaining high-quality recordings in a reasonably rapid manner. There is less time (and money) wasted waiting for the developer to respond to questions regarding pronunciation.

IP Glossary

After having worked on the original *Mass Effect* title, the *Mass Effect* 2 writers and editors also compiled a series of documents detailing the IP. These documents, together with our in-game *Codex* information, allowed us to create an encyclopedic compendium of IP terms, descriptions and pictures. We were able to detail all pertinent elements of the *Mass Effect* universe — from the history of the universe, to the various races that inhabit it — and provide this information to the localization team.

We found that this IP Glossary was useful for all external partners, and new members of the team — not just localization partners. This documentation allowed us to preserve cohesion throughout the *Mass Effect* universe, and anyone even tangentially-related to the project would be able to use the information contained in this series of documents to become familiar with the IP.

Translator Q&A Documents

One of the many enjoyable aspects of working on a sequel is the opportunity to re-use work that went into the previous game(s). The localization team maintains a document detailing translator questions, and the answers thereto. Wherever possible, we try to use the same translators for sequels, so that we can make use of their experience. That said, the possibility exists that they could need a refresher on the material — or that we could have to train a new translation team.

For *Mass Effect* 2 we were able to re-use the Q&A document from *Mass Effect* to ensure that, at the very least, translators didn't have to ask the same questions repeatedly. It makes little sense for translators to wait for us to answer questions that have already been answered; time that is spent waiting, rather than translating, is wasted time.

Character Bible

For actor casting purposes, both the EN VO team and the localization team make use of what we call a character bible. The character bible contains information that is not part of in-game data, so it does not appear in dialogues or menus. Rather, this metadata is used to describe the virtual characters experienced in-game. It contains pertinent background information such as name, gender, species, age, character archetype, importance (major/minor character), speech patterns, accent, demeanor, etc.

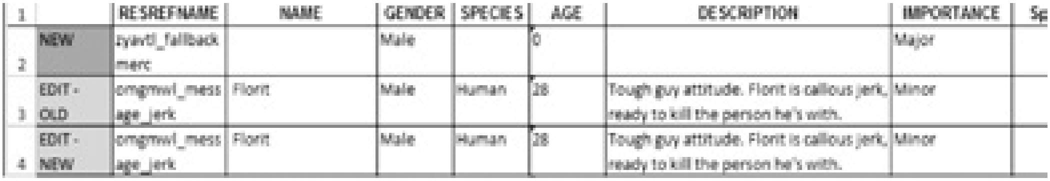
The character bible contains any information that an actor will need in order to accurately portray the character, as envisioned by the writers. As changes (additions, updates and deletions) are made to the characters over the course of development, delta (updated) information is tracked and displayed.

This information is not just useful to the VO recording teams, but it is also essential for translators as reference text. This is yet another way that the development and localization teams can share context with translators; we can reduce basic translator questions, which speeds up the translation process. Plus, it has the added benefit of enabling a higher-quality translation.

All this information must be tracked and displayed in an informative and unobtrusive manner, throughout the entire development process. *Mass Effect* 2 featured 572 distinct characters, and managing all of this data can become unwieldy. We needed a way to be able to locate specific characters at a glance — or at least to filter for characters quickly.

To that end, we have designed our character bible to be readable by Microsoft Excel. Excel is effectively a ubiquitous software package and, as such, allows any external partner to read our character bible. Excel's searching and filtering capabilities are particularly useful for quickly sifting through the reams of character information.

We have created two different versions of the character bible: one that is exported from our database, in a file format readable by Excel (as



1		RESREFNAME	NAME	GENDER	SPECIES	AGE	DESCRIPTION	IMPORTANCE	Sp
2	NEW	zyavtl_fallback merc		Male		0		Major	
3	EDIT - OLD	omgmwl_message_jerk	Florit	Male	Human	28	Tough guy attitude. Florit is callous jerk, ready to kill the person he's with.	Minor	
4	EDIT - NEW	omgmwl_message_jerk	Florit	Male	Human	28	Tough guy attitude. Florit is callous jerk, ready to kill the person he's with.	Minor	

shown above), and one that is a live snapshot of our database. The former includes more specificity, and is intended to accompany each batch of text sent for translation. The latter is a streamlined web report intended for use with external partners who wish to have access to up-to-the-minute information, without having to wait for a full text export.

Scheduling

If we had the luxury of working with a completed game, localization would be relatively simple. We could send all text out for translation in a single batch, receive translations in a few months, schedule and record some actors, and breeze through testing. Unfortunately, the reality of the global marketplace means that that is simply not an option. Localizing a game in the midst of development is significantly more difficult — especially where scheduling is concerned.

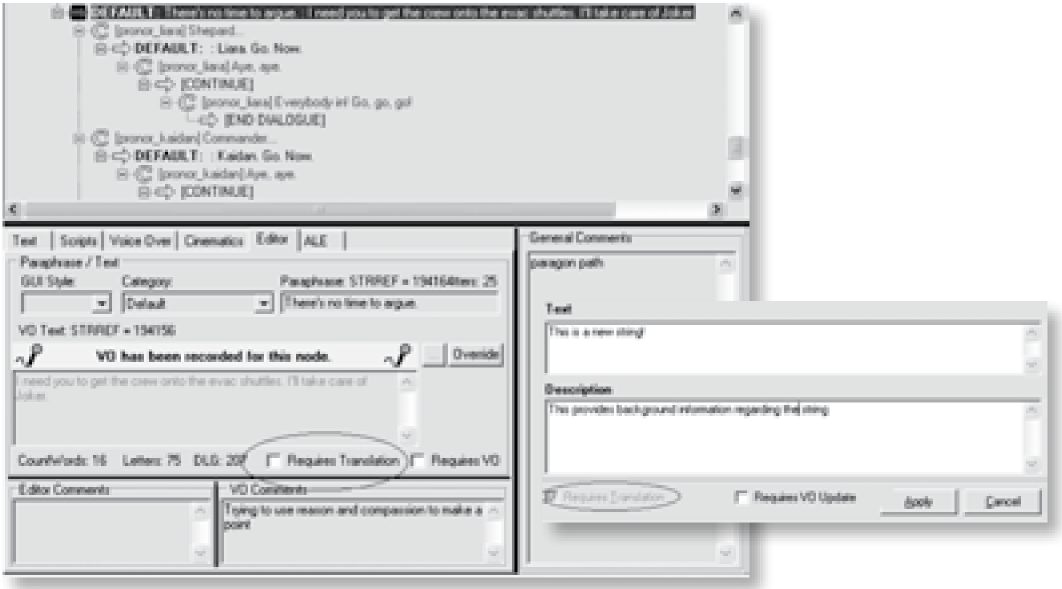
During the early stages of *Mass Effect* 2 development, the localization team consulted with the writing team often. An extensive and specific schedule was created. This schedule detailed which levels of the game would be ready for translation at any given time. As with any creative endeavor though, the monolithic and granular schedule quickly fell out-of-sync with reality. Levels had to be reworked, which quickly exhausted buffer time built into the schedule. It was the beginning of the localization phase of the project, and already the schedule was out of date.

The solution to the schedule problem was seemingly obvious: translation order should follow the order of EN VO recording, and the localization schedule should likewise follow the EN VO recording schedule — albeit delayed by approximately one month.

VO recording is expensive, regardless of the language being recorded. EN VO recording had to precede localized VO recording, so that reference audio would be available, and text could be considered reasonably final (and thus ready for translation).

It is possible for source text to be modified for the purposes of in-studio edits. Because subtitles can be enabled in the final game, subtitled text has to match the VO recorded by an actor. If an EN VO actor is more comfortable saying «the earth» instead of «the world», for example, the EN source text will have to be changed to reflect this altered phrasing. This alteration is considered a minor edit and does not impact localization, because the contextual meaning of the altered phrasing remains the same in both circumstances, even though the precise terms have differed, i.e., «le monde» can remain «le monde» in French.

More destructive in-studio edits, such as fundamentally altering the meaning of a line, can be disruptive to localization, but because our localization followed the EN VO recording by a month's worth of lead-time this was not an issue. The EN VO recording team had time to finalize in-studio edits, and build up a reasonable backlog of final content by the time



DEFAULT: There's no time to argue. I need you to get the crew onto the evac shuttles. I'll take care of Joker.
[pronoc_liara] Shepard...
DEFAULT: : Liara. Go. Now.
[pronoc_liara] Aye, aye.
[CONTINUE]
[pronoc_liara] Everybody in! Go, go, go!
[END DIALOGUE]
[pronoc_kaidan] Commander...
DEFAULT: : Kaidan. Go. Now.
[pronoc_kaidan] Aye, aye.
[CONTINUE]

Paraphrase / Text
GUI Style:
Category: Default
Paraphrase: STRREF = 194164 letters: 25 There's no time to argue.
VO Text: STRREF = 194156
VO has been recorded for this node.
I need you to get the crew onto the evac shuttles. I'll take care of Joker.
CountWords: 16 Letters: 75 DLG: 20 □ Requires Translation □ Requires VO
Editor Comments
VO Comments: Trying to use reason and compassion to make a point
General Comments: paragon path

Text: This is a new string!
Description: This provides background information regarding the string
☑ Requires Translation □ Requires VO Update

that localization needed to begin for the given in-game asset.

Once scheduling has been sorted out, the localization team follows a regular cycle of exporting localizable assets, translation, importing translations, printing VO scripts (as necessary) and recording VO (as necessary).

Translation

Major and Minor Edits

Text modifications are what trigger our approach to our entire localization process. Since this is driven by game data, we have programmed some localization intelligence into the custom applications used by our game designers and writers.

Our text is stored in a Microsoft SQL Server-powered database, and each of our content creation tools (the custom applications mentioned above) handles changes to this text in a consistent manner. Each text change is conceptually referred to as either a major text change/edit, or a minor text change/edit. Major edits are automatically marked as requiring translation, whereas minor edits are not. As individual designers or writers are likely to have different interpretations as to whether a text change should be considered major or minor, we work with our project leads to ensure that a single interpretation is used consistently throughout a project, i.e., leads are in charge of maintaining consistency.

We err on the side of caution, and initially flag all string changes (including when strings are created, as well as when existing strings are updated) as major edits. A string's major edit status is reflected in the content creation tools by a *Requires Translation* checkbox, which is set to «true» by default.

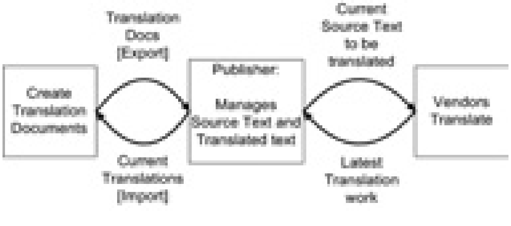
Shown above is a sampling of our conversation editor, and below our more general string editor, with the available *Requires Translation* checkbox. Writers and editors can manually override this setting when making string edits — as text and punctuation edits that do not adjust the meaning of a string are minor edits, and should not require translation. Though this decision is initially left to the discretion of writers and editors, our localization team has a specialized tool that can override this setting yet again, in conjunction with the localization specifications of our external partners. This doubly-redundant system is invaluable in preventing the unnecessary translation (and therefore expense) of thousands of strings.

Given the typical size of our projects, the translation pipeline already contains a high volume of text (and text changes). Only strings that have been marked as requiring translation are sent into the translation pipeline; as an efficiency measure, we must reduce the amount of «noise» in the process. Every text change sent to translators has to be reviewed and compared against existing translations on a per-language basis. It is a waste of resources to perform reviews only to find that a missing period was added, or that extra whitespace was removed from a portion of text, so we automate these checks before sending batches out to translators.

Exports vs. Imports

Standard Pipeline

Another fundamental concept that drives our localization process is the export and import portion of the translation pipeline.


Translation Docs [Export]
Current Source Text to be translated
Create Translation Documents
Publisher:
Manages Source Text and Translated text
Vendors Translate
Current Translations [Import]
Latest Translation work

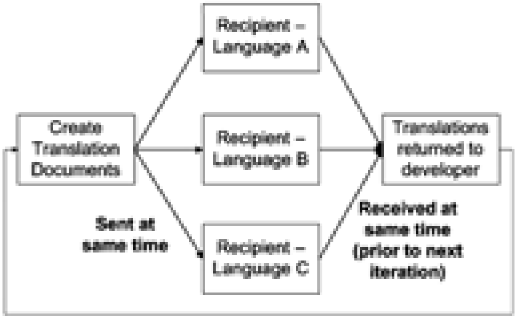
In brief, 'export' means that we are sending strings from our database for translation, and 'import' means that we have received translated strings for insertion into our database. These strings are then used to build into the game.

At BioWare, the translation pipeline generally follows the pattern illustrated above. Our software creates documents intended for translation, containing the most recent English source text. These are packaged and sent to the project's publisher. The publisher imports these documents into translation management software, and manages the translation process between their software and various translation vendors. This allows both source text and translations to be updated independently, with no need to ensure that an update to either be propagated immediately to the other. Updates to the source text will be propagated to the translation management software during the next export, and updates to the translations will be propagated from the translation management software during the next — thus avoiding the need to micromanage string translations.

A key benefit of having an intermediary translation management system is that it allows both processes to run concurrently. We can provide English source text updates at any time, and translators can provide new (or updated) translations whenever needed. Despite the frequency of updates to either source or translated text, there is a negligible risk of overwriting newer text with older data.

As before, English source text is meant to be as final as possible prior to its translation. There exists the possibility though, that it will change at some point. Because both the source and translated text are in states of possible flux, our tools have been programmed with the safest assumption, that translation exports and translation imports will happen independently; there may not be a one-to-one correlation.

Modified Pipeline


Recipient – Language A
Create Translation Documents
Recipient – Language B
Translations returned to developer
Sent at same time
Recipient – Language C
Received at same time (prior to next iteration)

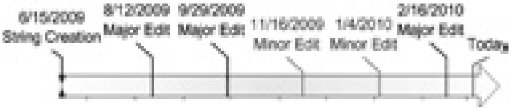
Some translation pipelines do not have an intermediate management system. In this case, documents may be sent to multiple recipients — making concurrent updates of English source text and translated text difficult. Because concurrency cannot be guaranteed, each document must be tracked separately. When using a modified pipeline such as this, we have found the need to keep each recipient in sync, ensuring that all parties send and receive documents at the same time. While such an endeavor can be organized well enough to function, micromanagement on a per-string basis is often required. Considering the scope of the average BioWare project, such micromanagement is not ideal.

Exports

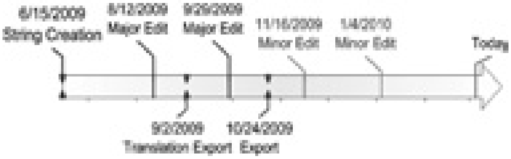
Each time the localization team performs a translation export, the translation pipeline must determine what text should be exported.

Our localization tool can represent each string as a separate timeline. When major and minor edits take place, they can be marked on the timeline as events. When a string is sent for translation, an export event can also be marked on its timeline. If translation exports were tracked separately for each given language, these events could also be marked on a string's timeline. To reduce the complexity of this tracking, given the amount of data involved, our pipelines treat each export event as an export to all applicable languages.

Most BioWare projects generate a large volume of text, and each string is revised many times. To this end, a database is used for its ability to manage and track all of these events, across all of the text, at the same time.

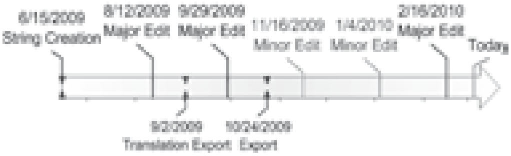


At any point in time, the strings requiring translation are considered as such when their timeline shows a major edit has occurred more recently than the last export. If a string has never been exported (as illustrated above) it will require translation by default; a major edit will always have occurred more recently than the last export.



As of the export on October 24, 2009, the string read «The human asked you to leave!!!» The minor edit on November 16, 2009 changed the string to read «The human asked you to leave!» and the minor edit on January 4, 2010 further changed the string to read «The human asked you to leave.»

These are considered minor edits, as they change only the punctuation and not the meaning of the sentence.



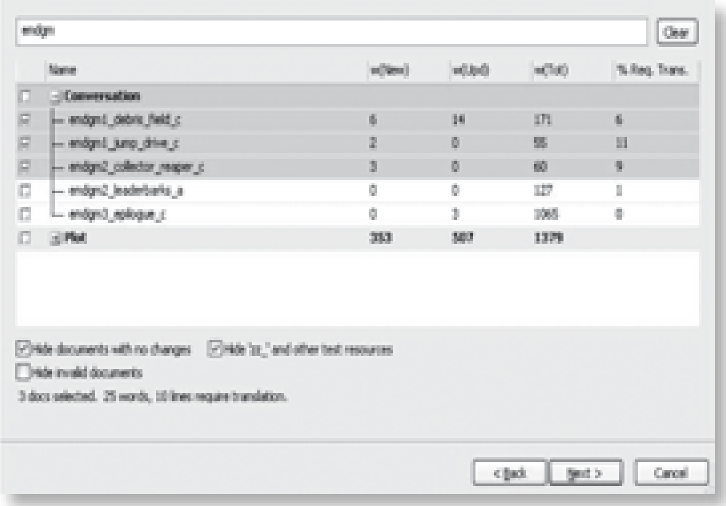
The major edit on February 16, 2010 changed the string to read «Shepard asked you to leave.» This is considered a major edit, since it does change the meaning of the sentence.

In the first example, the string was last exported on October 24, 2009 and has had only minor edits, so it does not require re-translation, and will not be exported. The second string was also last exported on October 24, 2009 but it has had a major edit since then, which means that it does require re-translation, and will be exported by the localization pipeline.

By broadly-applying this technique across the database, the localization pipeline can identify all strings that require translation. As an added benefit, we can create some useful statistics from this data.

The word and line counts of the resultant text provide information as to the volume of translation work necessary. The above image illustrates the English source word count for text requiring translation (as well as text requiring re-translation). This information is typically-used to determine how much source text will be sent to translators in a given delivery, since the amount of source text to be translated should closely match the throughput of the translation team. As an added benefit, any source text that is withheld from a given delivery can receive additional modifications without necessitating translation re-work.

Another useful metric gathered at this point is the amount of source text that will be sent for re-translation. The term «update» is used to describe this circumstance. The localization team can track the potentially-unwanted iteration on text, and translation costs can be estimated based on «fuzzy» (approximate) text matches.



endgm

	Name	w(New)	w(Upd)	w(Tot)	% Req. Trans.
□	Conversation				
☑	endgm1_debris_field_c	6	14	171	6
☑	endgm1_jump_drive_c	2	0	55	11
☑	endgm2_collector_reaper_c	3	0	60	9
□	endgm2_leaderbarks_a	0	0	127	1
□	endgm3_epilogue_c	0	3	1065	0
□	Plot	353	507	1379	

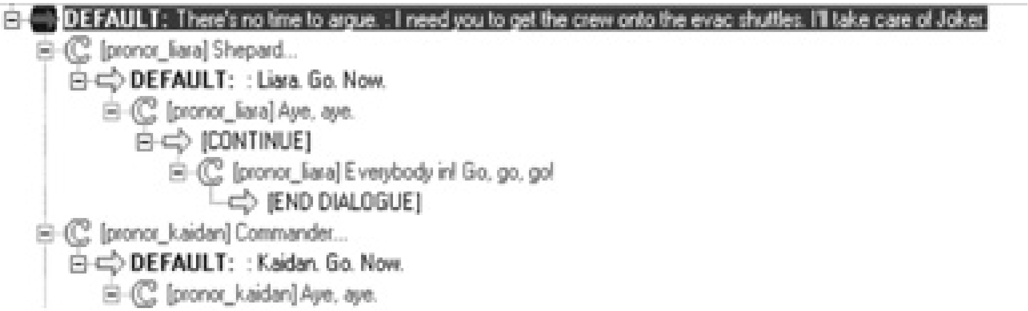
☑ Hide documents with no changes ☑ Hide 'zz_' and other test resources
□ Hide invalid documents

3 docs selected. 25 words, 10 lines require translation.

These statistics help manage a project regardless of its scale. Having insight into the overall translation effort needed helps the localization team forecast how many translation batches are necessary on a project — or, at least how many are necessary in the near future.

Non-Conversation Strings

So far we have identified which strings need to be translated, but further classification is necessary. If a random collection of text



```
DEFAULT: There's no time to argue. : I need you to get the crew onto the evac shuttles. I'll take care of Joker.
  [pronor_liara] Shepard...
    DEFAULT: : Liara. Go. Now.
      [pronor_liara] Aye, aye.
        [CONTINUE]
          [pronor_liara] Everybody in! Go, go, go!
            [END DIALOGUE]
  [pronor_kaidan] Commander...
    DEFAULT: : Kaidan. Go. Now.
      [pronor_kaidan] Aye, aye.
```

from a BioWare project was given to a translator, it would likely result in hundreds of questions and ultimately in a poor translation. Any given selection of English source text would not make much sense, and the translator could only guess at the context surrounding any given string. To improve translation quality, we want to provide as much contextual information as possible with every string.

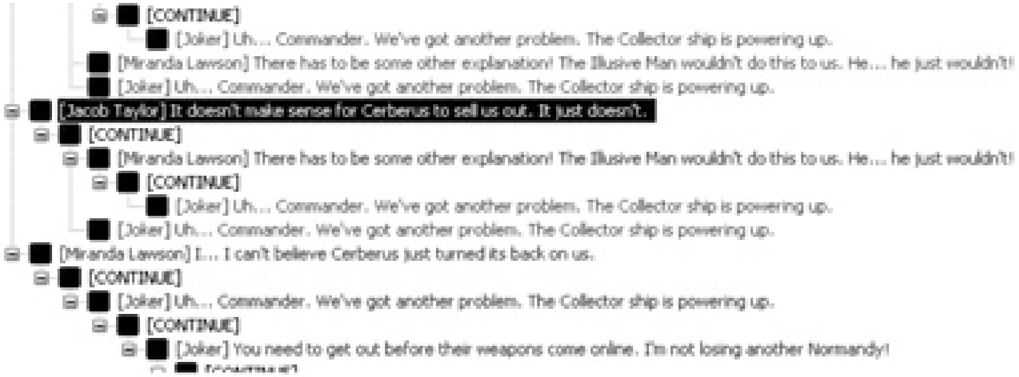
Every string that enters the database, for our projects, is branded with a resource 'parent', though there are some that do not logically belong to a sole resource parent (for example, on-screen user interface text). For this reason, strings are also branded with a *string type* — a representation/collection of similarly-handled text. Some examples of string types are:

- Achievements
- Art placeables
- Character names
- 360 GUI (Graphic User Interface)
- PC GUI
- Credits
- Error messages
- Galaxy map
- Loading hints
- Quest title, description

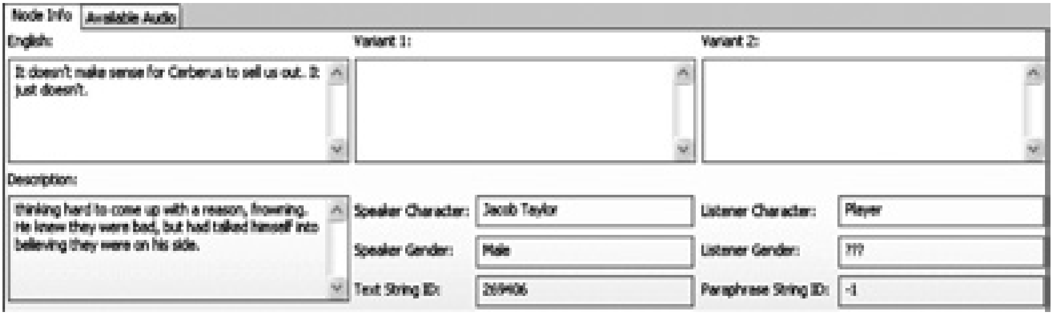
When preparing text to export for translation, it makes the most sense for the localization pipeline to group, provide context for, and package text based on its most distinguishable characteristics. Given the story-driven nature of BioWare's projects, the majority of in-game text belongs to conversation and journal resources — both *Mass Effect* 2 and *Dragon Age: Origins* contained roughly 80% of the total word count in dialogues, and roughly 10% in journal entries. The remaining 10% covered over twenty other string types. To that end, we decided to handle text from conversation and journal resources in a special manner.

Conversation Strings

Conversations that occur in BioWare titles contain many branching options, and are best-represented by a tree structure — since it is the player who is in control of the dialog. The order in which strings are presented to translators can be useful, for contextual purposes, but would require a linear structure. In order to harmonize this apparent conflict, the localization pipeline arranges the text from these conversation trees in a manner that makes sense to a human reader. It provides information about the tree structure of the conversation; each line indicates the line that preceded it, and which lines follow it. Translators are thus able to cross-reference adjacent lines of dialog.



```
        [CONTINUE]
            [Joker] Uh… Commander. We've got another problem. The Collector ship is powering up.
      [Miranda Lawson] There has to be some other explanation! The Illusive Man wouldn't do this to us. He… he just wouldn't!
      [Joker] Uh… Commander. We've got another problem. The Collector ship is powering up.
[Jacob Taylor] It doesn't make sense for Cerberus to sell us out. It just doesn't.
  [CONTINUE]
      [Miranda Lawson] There has to be some other explanation! The Illusive Man wouldn't do this to us. He… he just wouldn't!
        [CONTINUE]
            [Joker] Uh… Commander. We've got another problem. The Collector ship is powering up.
      [Joker] Uh… Commander. We've got another problem. The Collector ship is powering up.
[Miranda Lawson] I… I can't believe Cerberus just turned its back on us.
  [CONTINUE]
      [Joker] Uh… Commander. We've got another problem. The Collector ship is powering up.
        [CONTINUE]
            [Joker] You need to get out before their weapons come online. I'm not losing another Normandy!
              [CONTINUE]
```



English: It doesn't make sense for Cerberus to sell us out. It just doesn't.

Variant 1:

Variant 2:

Description: thinking hard to come up with a reason, frowning. He knew they were bad, but had talked himself into believing they were on his side.

Speaker Character: Jacob Taylor

Speaker Gender: Male

Text String ID: 269406

Listener Character: Player

Listener Gender: ???

Paraphrase String ID: -1

This conversation structure is so important that we provide translators with a tool called the «Conversation Previewer», since translators do not have access to content creation tools like the conversation editor that is used by our designers and writers. The Conversation Previewer therefore provides a way for translators to visualize conversation structures in a more natural manner very similar to the structure used in the content creation tools.

The Conversation Previewer presents the English source text, and its available translations and localized VO files, as an accurate representation of how translated content will be experienced in-game, as well as how the English source text flows.

Conversation Meta-data

The game engine is aware of which characters speak each conversation line and, in most cases, the game is aware of which characters are listening to the speaker. This information can be combined with the character bible, and the localization pipeline can provide translators with the speaker and listener character information for each string. Early validation checks are made, and any conversation containing ambiguous, or undefined, speaker information is withheld from translation until character associations can be made for each line. As a result, the number of questions sent back from translators regarding the source conversation text, has been greatly reduced since this conversation metadata gives precise information on gender, number, status, etc.

Timing and length restriction are another piece of information that is sent along with conversation text. Lines that require VO recording, and which will be matched to animations of a fixed length, are flagged. This flag indicates that the length of the translated text should closely match the length of the source text. While such restrictions are not ideal, they are sometimes necessary due to certain core game mechanics.

Data Medium

There is a multitude of information that is imperative to present to translators on a per-string basis, and it is therefore important that all of this data remain together through the localization pipeline. We have found that XML format works well as a medium for sending this data. XML files provide a conduit for clearly-defining all contextual information for each string. Since these XML files are typically consumed by another program before the data is presented to human translators, the use of XML is a flexible alternative to proprietary or custom binary file formats.

Many translation vendors use SDLX-based ITD files as a transfer medium, which makes for an attractive medium due to such widespread adoption. Unfortunately ITD files do not easily display the meta-data that is provided by our localization pipeline, and they become exponentially slower as we add the volume of text typical to a BioWare project. XML files are substantially faster to create and process.

Another benefit of XML files is that our use of a consistent structure has allowed us to create cross-project applications. These applications do not require specialized changes, or a separate codebase, for each project — and are therefore easier to maintain. The Conversation Previewer, as mentioned above, is one such example.

Validation

Validation checks are performed prior to import to ensure that the translations being imported will not pose any problems in future game builds. The localization pipeline import software has further checks that act as reactionary measures when bad data somehow avoids the validation checks. This bad data must not be propagated into the game data, but it can be useful in identifying potential process gaps. If we can identify how the bad data originated, we can prevent it from occurring. The end result is an automated system that ensures that such issues will not occur as easily in the future.

Validation checks can be warning-based allowing an import process to proceed, or they can be error-based and halt the import process. An example of a non-critical warning would be the identification of empty (or nonexistent) translations. This isn't optimal, but the import process can still continue without threat to the game content. In another scenario, we might receive multiple translations for a single string. We would want to prevent any translations in this delivery from entering the database until the cause of the error has been identified, and solved.

On some projects, these validation checks have been spun off into a standalone validation tool. The publisher or translation vendor can use this tool directly, reducing the turnaround time incurred when bad data is isolated. Each turnaround would typically require at least an additional day, due to time zone constraints.

Localized VO Recording

Once translations have been imported into the database, the localization pipeline allows for the printing of localized VO recording scripts. These scripts have been optimized such that they allow for contextual organization, as well as ease-of-reading, and the scripts can be generated with same-day turnaround time. Because the translation process has followed EN VO recording by a month, by the time that the localization team is ready to print localized VO recording scripts, EN reference audio is available for bundling with the associated scripts.

There are two different types of scripts, for each character: actor scripts, and audio scripts. Actor scripts tend to be the most relevant one of the two, for localization. These scripts include unique line numbering (for ease of reference), the EN source text, translated text, director comments (in English), and other contextual information that may help with managing the sheer volume of scripts that are generated.

Lines that are to be recorded in a particular session are bolded, and are surrounded by other characters' lines, to preserve context for the actor. If a line is not bolded, yet is meant to be spoken by the character being recorded, it indicates that the line was recorded in a prior session, or earlier in the same script.

File naming conventions are also included, on a per-line basis. This allows the studio to cut and name the lines such that they meet BioWare's naming conventions. Deviation from the naming convention assures that the file will not play in-game, so this is important information for the recording studio/mastering lab.

CONCLUSION

With adaptations over the years, this represents the evolution of the localization process BioWare has employed on projects for the last seven years. *Sonic Chronicles* on the Nintendo DS and the upcoming MMO RPG *Star Wars: the Old Republic* have a variation in scope of hundreds of thousands of words. The localization tools and processes we have developed over the years enable us to tackle both projects with equal confidence. Players all over the world remain immersed and in control of their game experience because the localized version is as immersive as the original.

RECIBIDO Y VERSIÓN FINAL: OCTUBRE DE 2010
ACEPTADO EN ENERO DE 2011