Translation Memory Mode Concepts Guide

SDL WorldServer 10.4.3

Documentation Notice

Copyright and trademark information relating to this product release

This product may include open source or similar third-party software, details of which can be found by clicking the following link: *“Acknowledgments”* .

Although SDL Group takes all reasonable measures to provide accurate and comprehensive information about the product, this information is provided as-is and all warranties, conditions or other terms concerning the documentation whether express or implied by statute, common law or otherwise (including those relating to satisfactory quality and fitness for purposes) are excluded to the extent permitted by law.

To the maximum extent permitted by law, SDL Group shall not be liable in contract, tort (including negligence or breach of statutory duty) or otherwise for any loss, injury, claim liability or damage of any kind or arising out of, or in connection with, the use or performance of the Software Documentation even if such losses and/or damages were foreseen, foreseeable or known, for: (a) loss of, damage to or corruption of data, (b) economic loss, (c) loss of actual or anticipated profits, (d) loss of business revenue, (e) loss of anticipated savings, (f) loss of business, (g) loss of opportunity, (h) loss of goodwill, or (i) any indirect, special, incidental or consequential loss or damage howsoever caused.

November 2014

Contents

1 About the Translation Memory Mode Concepts Guide 1

SDL WorldServer Documentation and Resources 1

2 Translation Memory Modes 5

TM Mode Comparison 5

Key Concepts 6

Scoping Leveraging and Scoring 7

Tokens and Word Counts in Studio 8

Repair Comparison 8

Penalties in Studio-Aligned Mode 9

Repetition Analysis and Scoring 11

Auto Translation and Scoring 11

Relaxed Tag Parity 12

Disabled Customizations 13

Live Translation Memory Mode 14

Differences Between Live TM Mode and Non-Live Mode 16

Migrating to Live Translation Memory Mode 18

3 TM Mode Implementation 21

TM Mode Settings 21

Project Settings Changes in Studio 22

Auto Translation in Studio-Aligned Mode 22

User Interface Changes 23

Real-Time Studio Updates 24

4 Studio Examples 25

Recognized Token Example 25

Markup Tags Example 27

Chapter 1

About the Translation Memory Mode Concepts Guide

Welcome to SDL Language Technologies and WorldServer™. SDL WorldServer is a translation management system that provides advanced linguistic technology, process automation, content repository integration and management services. Aligned with SDL Studio™, WorldServer provides consistent analysis and reporting of translation projects for localization managers, project managers, translators and reviewers.

Scope

The *SDL WorldServer Translation Memory Mode Concepts Guide* provides information on the Studio-aligned translation memory (TM) mode introduced with WorldServer 2011. It is designed for advanced language administrators and project managers who need to understand how WorldServer supports TM concepts (such as tokens, penalties, and leveraging) for the SDL Trados Studio environment.

SDL WorldServer Documentation and Resources

The SDL WorldServer documentation set includes multiple separate deliverables. Some of these documents are installed with WorldServer.

The following publications are available from the SDL Knowledgebase, in the Product Documentation/SDL_WorldServer folder.

Administrator Guide

Information on setting up and administering WorldServer in your environment.

Administrator Tutorial - Configuring SDL WorldServer

The process of performing an initial configuration of WorldServer.

Basic Operations Guide

Basic system administration and troubleshooting information for installed WorldServer systems.

Browser Workbench User Guide

Information on embedded translation tool for brief reviews or edits.

Installation and Upgrade Guide

Instructions for installing or upgrading WorldServer and its components in your environment.

Planning and Prerequisites Guide

Conceptual information for planning a WorldServer installation or upgrade, including environment requirements and options for the WorldServer platform, infrastructure, and supported components.

SDK Web Services Developer Guide

Reference guide for programmers who need to add Web Services to WorldServer.

Software Development Kit (SDK) User Guide

Reference guide for programmers to understand the APIs that extend WorldServer functionality and integrate WorldServer with other custom or third-party applications.

Translation Memory Mode Concepts Guide

Information for project managers that need to understand the Studio-aligned translation memory mode.

User Guide

Information for general WorldServer users: project managers, translators, and reviewers.

WorldServer and Studio Integration Guide

Information for translators, reviewers, and project managers that use SDL Studio with WorldServer.

The following WorldServer documentation resources are also available:

WorldServer Release Notes

Where: The Knowledgebase SDL WorldServer/Release Notes folder. Also available from the WorldServer FTP site: `Walthamftp.sdl.com/ dist/ product/ worldserver/<releasenumber>/`

Information on enhancements and changes in the most recent release. Also contains late-breaking information about known issues and issues fixed in this release.

SDL Glossary

Where: http://producthelp.sdl.com/glossary/en/glossary.htm

An alphabetical list of terms related to WorldServer and other SDL products.

Transport Online Help

Context-sensitive help is available for most pages in TransPort from the **More Info...** link. A full web help system is available from the **Help** link at the top of each page.

All Language Technologies online help documentation is also available from the following web page: `“http://www.sdl.com/support/language-technology-products/trados-online-product-help.html”http://www.sdl.com/support/language-technology-products/trados-online-product-help.html/`

WorldServer Online Help

Context-sensitive help is available for most pages in WorldServer from the **More Info...** link. A full web help system is available from the **Help** link at the top of each page.

Chapter 2

Translation Memory Modes

Starting with SDL WorldServer 2011 (WorldServer version 10.0), new installations will only use the Studio-aligned TM mode to create projects. The Studio-aligned mode applies translation memory (TM) to projects in the same way as SDL Trados Studio.

When you work in the Studio-aligned TM mode, note the following:

- For best results, use the Studio-aligned TM mode with the File Type Support (FTS) server.
- The studio-aligned TM mode does not automatically switch to FTS file types. You must enable the file types you need separately.
- Always use the most recent version of Studio with WorldServer 2011.

Existing WorldServer customers may choose either the Studio-aligned TM mode or the WorldServer 9.x TM mode when they create projects. This decision depends on whether you want to integrate your projects with the SDL Studio translation supply-chain. Once that choice is made, it remains in force for the life of the project.

Note: You use the same translation memory in either mode. It's just how WorldServer applies the TM that changes.

TM Mode Comparison

TM modes can affect pre-translation work in WorldServer during the following operations:

- Scoping assets (assets such as files, database entries, etc. are translatable items in Studio terminology)
- Opening translatable items in Browser Workbench
- Running projects in a workflow
- Searching a translation memory with the **Standard** search type.

Studio-aligned TM mode

Going forward, WorldServer is focusing on supply chain integration. SDL Trados Studio is the dominant tool used by professional translators, so much so that many third party tools base their features on compatibility with Studio.

Users should choose the Studio-aligned mode if they want to standardize their translation supply chain interactions. For example, this mode provides more consistency for word counts and leverage when sending projects to language service providers (LSPs) that use Studio. Because this is the future direction of WorldServer, users should use the new File Type Support (FTS) filters and Studio-aligned mode. This will ensure maximum consistency between WorldServer and Studio.

WorldServer 9.x TM mode

As the name implies, the 9.x mode is backwards-compatible mode with previous versions of WorldServer. Existing WorldServer customers may not be able to transition all their work to the Studio-aligned TM mode immediately.

Users should use the 9.x mode when their primary concern is the consistent use of an existing TM for documents that have been translated in WorldServer before the 10.0 release. If users are sending modified documents through the system, using legacy filters, and applying existing TM, the 9.x mode provides maximum leverage. However, that maximized leverage will be at the cost of the Studio-specific capabilities and supply chain integration features offered in the Studio-aligned mode.

Users who decide to stay with the 9.x mode for now, should have a plan for changing to the Studio-aligned mode in the future.

Key Concepts

After WorldServer performs segmentation on selected assets (also called translatable items), the Studio-aligned translation memory (TM) mode generally matches the behavior of SDL Trados Studio in the following areas:

- Scoping and leveraging
- Tokens and word counts
- Penalties and repair
- Repetition analysis and scoring
- Relaxed tag parity

Scoping Leveraging and Scoring

The differences in the two WorldServer translation memory (TM) modes are perhaps most apparent when you scope a project. WorldServer 9.x mode and the Studio-aligned mode use very different underlying technologies to break assets (translatable items) into segments, and to break segments into tokens. The Studio-aligned mode produces the same scores as if you scoped the work in Studio itself. SDL Trados Studio users will appreciate this consistency.

We anticipate that users will choose to do all their work in one mode or the other. If you compare the scores that result from scoping the same assets using different TM modes, the scores for the translation matches will differ, producing different leverage levels. If customer cost models depend on leverage levels, the costs associated with the projects could change if you change TM modes.

Language Fallback

In SDL Trados Studio, languages without specified regions (so-called *"region-neutral cultures"* or *"region-neutral languages/locales"* such as `en` or `de`) are not supported. Instead, users must use a fully region-qualified language (*`en-US`*, *`en-GB`*, *`de-CH`*, …) to obtain the correct date, time, and number patterns. For that reason, WorldServer internally now uses the most likely region-qualified language in case a WorldServer language doesn't specify one. For example, *`en-US`* will be used for *`en`*, and *`fr-FR`* will be used for *`fr`*.

Note: The fallback is applied only if users use a region-neutral locale. We strongly recommend specifying a region-qualified locale to avoid implicit fallbacks.

WorldServer and Studio TM Engines May Return Different Translation Units

The Studio-aligned TM mode makes sure that the same translation unit returned by the TM engine gets the same score as it would in SDL Trados Studio. However, the two TM engines may return different translation units (TUs) as the "winning" match for some document (lookup) segments. If that happens, the two returned TUs may be scored differently, which may result in minor differences in the scoping reports if the two TU scores end up in different scoping bands.

This is expected to occur only rarely, and the likelihood is higher in the lower-score fuzzy bands. For higher fuzzy bands, this should occur very infrequently.

In very rare circumstances, it may even be that no TU will be returned by Studio, in which case the document lookup segment will be reported as a "no match" (new translation) against the TM in Studio.

Tokens and Word Counts in Studio

The Studio-aligned TM mode can automatically substitute recognized tokens (a piece of text in one of a pre-defined set of formats) for the text in a translatable item (file, asset, database entry, etc.) when WorldServer computes a translation proposal. The TM then recognizes and counts these tokens as a single word.

The following token types are recognized:

- Dates
- Times
- Numbers (in numerals)
- Measurements
- Acronyms and URLs

Note: Inline tags (extensions of the current WorldServer placeholders) are not counted as words.

This treatment can reduce the word counts (and hence costs) for some types of content. The Studio-aligned mode handles simple differences (such as international formatting) in these tokens in a way that results in better matches for translated information. For certain types of content this could enhance project processing performance and quality.

The Studio-aligned TM mode recognizes this special kind of content and auto-substitutes (replaces) appropriate recognized tokens. Recognition and related settings can be controlled through the `tm.properties` file.

Repair Comparison

The tables summarize how the WorldServer 9.x and Studio-aligned TM modes handle Studio and WorldServer repairs. Separate topics describe the TM mode differences in more detail.

Note: Auto-substituted means that tokens will replace the recognized format. The auto-localization step of auto-substitution means that the TM mode applies a specific replacement format to the recognized token, consistent with the target language, during auto-substitution. This process is also known as *repair*.

Table 1: Studio Repairs/Auto-substitution

Repair type	Studio-aligned Mode	WorldServer 9.x Mode
Numbers	Auto-localized.	Raw source form only (number transform repair), no auto-localization. Enabled by default.
Date and time expressions	Auto-localized.	Not supported.
Variables	Not applicable (future release).	Not supported.
Measurements	Auto-localized.	Not supported.
Tags	Auto-substitution is always enabled. Additional source tags are auto-copied to the translation proposal.	Auto-substitution can be disabled. Additional source tags are appended to translation proposal. Enabled by default.
Acronyms and URLs	Not auto-substituted, though they are treated as tokens. No effect on leverage.	Not supported.

Table 2: WorldServer Repairs/Auto-substitution

Repair type	Studio-aligned Mode	WorldServer 9.x Mode
Custom	Not supported.	Through customization.
Number deletion	Not supported.	Built-in, enabled by default.
Placeholder	Subset of tag auto-substitution.	Built-in, enabled by default.
Word transform	Not supported.	Built-in, enabled by default.
Word deletion	Not supported.	Built-in, enabled by default.
Punctuation	Not supported.	Built-in, enabled by default.
Capitalization	Not supported.	Built-in, enabled by default.

Penalties in Studio-Aligned Mode

In Studio-aligned mode, translation activity tries to make sure that the content of the translation, including the markup tags used in the translation, are as accurate as possible. You can control how penalties are set and which penalty values are applied. With the penalty settings, you can avoid assigning a 100% match score in situations where you want to ensure that the translation proposal is reviewed and treated as a fuzzy match.

Note: A combination of `tm.properties` settings and workflows determine when penalties are set. If the `tm.properties` entry for penalties is equal to zero (`=0.00`), penalties will not be set at all.

Penalties are primarily assessed for content and formatting inconsistencies. TM group penalties are applied as in the 9.x mode.

Content-based Penalties

If WorldServer automatically substitutes a piece of content during translation, it assesses a penalty on the translation unit. This penalty can be used to guarantee that (exact) matches produced by auto substitution are reviewed. The penalty is not triggered by tag-related auto substitutions. By default, the auto substitution penalty is set to zero (`=0.00`).

Metadata Penalty

The `tm_score_metadata_mismatch_penalty` setting in the `tm.properties` file extends 9.x mode behavior to the Studio-aligned mode. You have the option of applying the metadata mismatch penalty during WorldServer TM leverage when the metadata of the TM match does not match the metadata of the asset being translated. This is a global setti ng. The default is `0%`.

To enable this penalty, you must map TM metadata to asset metadata:

1. Create an AIS property with property type `Text`.
2. Create a TM Entry attribute with the internal name matching the AIS property name. Set its attribute type to `Text Field`.
3. Set `map_ais_property_to_same_named_tm_attribute= true` in the `tm.properties` file. The AIS property value will populate the value of TM attribute with the same name when the translation is saved.
4. Set `tm_score_metadata_mismatch_penalty=` `<number>` in the `tm.properties` file.
   <*number*> must be set to a number between 0 and 1. For example, `tm_score_metadata_mismatch_penalty=0.02`.

Formatting-related Penalties

There are two penalties related to formatting auto substitution and repair: the *missing formatting penalty* and the *different formatting penalty*.

- **Different formatting penalty.** Triggered if the markup tags in the translatable item source cannot be fully aligned with tags in the memory source segment.
- **Missing formatting penalty.** Triggered if the document source segment does not contain any tags, but the translation unit's source segment does. In this case, all tags are dropped from the translation proposal and the penalty will be set (note that the **different formatting penalty** will not be triggered in addition).

Note: The details of penalty-handling are defined in the `tm.properties` file. Penalties should be configured as floating point numbers with a value ranging from 0.01 to 1.00. The defaults match Studio penalties.

Note: Penalties are added to packages so that Studio can apply them consistently.

Repetition Analysis and Scoring

The Studio-aligned TM mode ensures that WorldServer handles repetitions in a way that duplicates SDL Studio 2011 repetition analysis.

Markup tags of the same type (singleton, opening, closing) appearing at the same location are considered equivalent for repetition analysis, as are segments which would be marked as 100% after repairs (auto substitutions).

The following token types (if enabled) are treated as interchangeable, and differences are considered equivalent in the repetition analysis, unless auto localization is disabled or the auto substitution penalty is set:

- Dates
- Times
- Measurements
- Numbers

The following segments are included in repetition analysis:

- Untranslated segments with no TM matches
- Segments for which there are only fuzzy matches (whether pre-translated or not)
- Manually translated segments

In order to assure scoring consistency, the following segments are excluded from repetition analysis:

- ICE matched segments
- 100% matched segments (with or without repair)
- Auto translated segments

Auto Translation and Scoring

In SDL Trados Studio 2011, once auto translation is applied, the segment status is set to machine translation. WorldServer maintains the distinction between auto translation and machine translation for scoping and scoring purposes.

Studio auto translation is a part of the translation memory (TM) leverage process. When no TM matches are found for the segment, auto translation rules go into effect.

The following segments can be auto-translated:

- Segments which *only* contain a number
- Segments which *only* contain a measurement
- Segments which *only* contain a URL

If applied, auto translated segments will get true auto-translated status. Number-only auto translations will be scored at 100%, and the other two cases will be scored at 95%, to enforce review.

The corresponding recognizers and auto localization features must be enabled in order for auto translation to work.

In a Studio project package, auto translated segments get **`Auto-Translation`** status, which Studio uses for both auto translated as well as machine translated segments

Note: Auto translated segments participate in the scoping results. What this means is that the segments are scored, and this score is used to attribute the words and segment to the corresponding scoping buckets. This is consistent with WorldServer.

Note: Machine translated segments in Studio generally do not participate in the scoping since their score is ignored. Auto translated segments are an exception to this rule, even though they have a machine translation status.

Note: In Studio-aligned mode, auto translated segments may not always be scored at 100%. WorldServer 9.x mode does score these segments at 100%.

In the WorldServer Studio-aligned TM mode, auto translation suggestions will continue to be incorporated in the standard fuzzy/score bands, just like TM matches. They will not be reported in the machine translation category.

When WorldServer project packages are imported into Studio, auto-translated proposals will be marked as `AT` in Studio, which is used by Studio to designate both auto translation as well as machine translation status. In Studio, these auto translation suggestions will be incorporated in the scoping reports (**Analysis Reports**) just like matches retrieved from the translation memory, with standard Studio scoring.

See the *TM Mode Implementation* chapter for scoring details.

Relaxed Tag Parity

Starting with SDL Trados Studio 2011 and SDL WorldServer 2011 (release 10.2.1), the default WorldServer environment (using Studio file types and Studio-aligned translation memory mode) relaxes the requirement for tag parity in translation unit (TU) segments. That is, some tags may be present in the translation target without a requirement that they be in the translation source, or vice versa.

Studio 2011 and 2014 create return packages for WorldServer 10.2.1 without enforcing tag parity. Things to note:

- Translation units which have tags that are out of parity are excluded from WorldServer in-context exact (ICE) matches.
- WorldServer resolves target-only tags through an automatic mechanism in the associated WorldServer filter configuration for the file type, in general. If the configuration does not use a Studio file type, or has another issue, leverage penalties will be applied.

Note: If you work in WorldServer's Browser Workbench, you may notice temporary non-sequential placeholder ID numbering for non-parity segments brought in from translation memory. The ID sequence generally resolves once a server round trip takes place for the segment (such as when you save an asset, or add a segment comment), if not before.

Legacy tag parity behaviors/settings

When you create a return package for a WorldServer version before 10.1, Studio 2014 and higher versions check for tag parity, the previous default behavior. The return package creation will fail in Studio if tag differences exist in that case. Importing an out-of-parity return package into pre-10.1 WorldServer environments will also fail.

If you do want to enforce tag parity in source and target translation units for a particular Studio file type, most file types have a **Tag Check** control that you can use to verify tag parity, the behavior prior to WorldServer 10.1.

Disabled Customizations

Studio-aligned TM mode does not support WorldServer 9.x TM mode customizations.

The following customizations are disabled:

- Customized stemmers
- Customized repair components
- Customized word breakers

These customizations are only enabled in WorldServer 9.x TM mode.

Live Translation Memory Mode

Translation memories can be configured to operate in *live* mode. In live mode, entries are added to the translation memory constantly during the translation process, and status is tracked to distinguish different types of entries. Translation memories are updated whenever the segmented asset cache is updated. Any operation that updates the segmented asset cache will update the translation memory (for example, saves to AIS, uploads from desktop translation tools, and so on).

In addition to enabling real-time sharing of translation memory, live translation memory makes it possible to effectively update projects by restarting tasks without losing work.

Note: To enable live mode, go to `tm.properties`, uncomment the line:

```
#enable_live_translation_memory=false
```

and change `false` to `true`. After you restart WorldServer, you will be in live mode. Live translation memory represents a substantial change from non-live TM mode. Existing customers may have to make conceptual, process, workflow and permission changes to adopt live mode. WorldServer default installations are set to non-live mode.

The *Translation Status* of a segment refers to if it is **Pending Review**, **Reviewed**, **Rejected**, or has no status.

Note: There is translation status on a *segment* (in the workbench and cache), and translation status on a *TM entry*, as described in the following table.

The combination of live translation memory mode and segment status makes it possible for translations to go into translation memory as soon as possible for the benefit of all translators, but at the same time translators can distinguish between translations that have been through a review process and those that might have just been added.

As translators work, the source and target text are copied from the segment cache to translation memory. Segment translation status determines if a segment is added to or updated in the translation memory and what translation status the corresponding entry should have. The following table shows what happens to the status in the translation memory when a segment's translation status changes in the workbench:

Table 3: Segment status in workbench and TM status

Segment translation status	Add or update made in TM?	TM entry translation status
None	No	No update
Pending Review	Yes	Pending Review
Reviewed	Yes	Reviewed

Segment translation status	Add or update made in TM?	TM entry translation status
Rejected	Yes	Rejected

When WorldServer first segments your file and creates the bilingual segment list, it tries to fill in some of the translations from the translation memory. If it finds an ICE match or a 100% match it will automatically fill them in. (Note that by default only TM entries with a status of **Reviewed** can result in ICE matches.)

If WorldServer finds an ICE match, it fills in the target segment with it and automatically sets the segment translation status to **Reviewed**. If WorldServer finds a 100% match, it fills in the target segment with it and by default automatically sets the segment review status to **Pending Review**.

Note: You can reconfigure how you want WorldServer to treat 100% matches. You can change it to set the status to **None** or to **Reviewed** (not recommended) by going to `tm.properties`, uncommenting the line:

```
#maximum_exact_translation_status=Pending
```

and changing `Pending` to `None` or `Reviewed`.

When a translator starts working, after she or he translates a segment and clicks **Save**, the translation goes into the translation memory as **Pending Review**. Later, if a reviewer changes the translation status of the segment in the segment cache (that is, in the workbench) to **Reviewed** and clicks **Save**, the corresponding entry in translation memory is also set to **Reviewed**.

By the end of the workflow, all segments should have a **Reviewed** status in the translation memory. When the reviewer sets the status of a segment to **Rejected**, the asset should go back to the translator. The translator can use the **All rejected** filter in the Browser Workbench to see only the rejected segments. When the translator fixes the rejected segment, he or she then changes the segment to **Pending Review** manually, using the **Tools** > **Translation Status** menu. When the translator has fixed the rejected segments, the asset goes back to the reviewer, and this cycle continues until all segments are marked **Reviewed** by the reviewer.

Additional TM Properties for Live Translation Memory Mode

In addition to the `enable_live_translation_memory` and `maximum_exact_translation_status` properties already described, there are two other settings in `tm.properties` with which you can control live translation memory mode behavior. These properties are commented out by default.

- `require_reviewed_status_for_ice= true` – Determines if translation entries must have **Reviewed** translation status as a requirement for ICE matches. The default behavior (which is, effectively, `true`) is that unreviewed translation memory entries do not satisfy the ICE criteria. To make **Pending Review** TM entries with ICE match scores have their segment translation status set to **Pending Review** (instead of **Reviewed**), uncomment this property and set it to `false`. In non-live mode, all translation memory entries are reviewed, so ICE matches are unaffected by this configuration.
- `tm_score_unreviewed_match_penalty= 0` – Defines the penalty for TM matches with an unreviewed translation status. In non-live mode, all translation memory entries are reviewed, so ICE matches are unaffected by this configuration. This is a leverage level penalty. It is applied to the final score produced by the match, and is only applied during leverage based processes. The value should be between 0 and 1. This penalty applies to all TM matches.

Differences Between Live TM Mode and Non-Live Mode

The following table summarizes some of the behavior differences between live translation memory mode and non-live mode.

Area	Live Mode On	Live Mode Off
TM Update	Whenever segment cache is touched.	On explicit calls to update TM. (Note that all entries added to TM are set to **Reviewed**.)
Browser Workbench	No **Save and Update TM** button.	Has a **Save and Update TM** button as in previous WorldServer versions.
XLZ Import	No **Update the translation memory using imported assets** check box.	Has an **Update the translation memory using imported assets** check box as in previous versions of WorldServer.
Import of other translation kit formats (besides XLZ)	Segments with a `none` status are set to **Pending Review**.	No special behavior.

Area	Live Mode On	Live Mode Off
User type permissions and ACL privileges	The following user type permissions are meaningless and hidden: - Can save asset and update TM - Can modify TM during Upload If a user that has only Read ACL privilege for a TM opens an asset in the Browser Workbench, the workbench does not present a **Save** button, and issues this message: `WARNING: You do not have permission to update translation memory "<TM>"`. Write permission is required when using Live Translation Memory. Please, contact your WorldServer Administrator. For the user to be able to work in the Browser Workbench in Live TM mode, the administrator must give that user Write ACL privilege for that TM.	Same as in previous versions of WorldServer.
Logging	Anytime `WSAssetTranslation. saveTranslationToTm()` is called, a warning is logged with a stack trace.	No special behavior.
Leverage	TM entry translation status affects segment status.	TM entry translation status does not affect segment status unless **Segment Asset** automatic action is used with `Set Translation Statuses?` and `Maximum Translation Status for 100% Matches` arguments set to `Yes` and `Pending`.

Migrating to Live Translation Memory Mode

When you upgrade an existing system to WorldServer 10.0 and enable live translation memory (TM) mode, you must be aware of a number of considerations.

Note: Live TM mode was introduced in WorldServer version 9.0.

When you upgrade to WorldServer 10.0, the status of all of your TM entries is set to **Reviewed**. Also, in non-live mode, whenever an entry is written to the translation memory it is given a reviewed status. This holds for all places that the translation memory is updated, including the Browser Workbench, import, and automatic actions such as `Save`. This helps you learn that entries added to translation memory the old way are considered “Reviewed” and also ensures that leverage works correctly when you first enable live mode.

Note: There is no need to specifically change leverage behavior, scoring, or concordance search for non-live mode, since every entry in the translation memory has a status of **Reviewed**. Any penalty or special behavior based on TM entries with non-Reviewed statuses therefore never occurs.

Because you might have workflow customizations that make explicit calls to update the TM from a segment cache, WorldServer checks for such explicit calls when live mode is enabled. If it detects a call to update the translation memory, it logs a warning indicating that the approach is deprecated. This is intended to help you find parts of your configuration that are not taking advantage of or are incompatible with live mode. As an example, you might have a `Save` automatic action with `Update TM` enabled.

Consider the following:

- In the old model, your translation memory was only updated when you specifically indicated to update the TM or if you put certain automatic actions such as "save” in your workflow. Your main concern was making sure that everything got into TM by the end of your projects. In the new model any segment with a translation status other than `none` will go into the translation memory. Your main concern is making sure that by the end of your workflow all segments are set to **Reviewed**. Look at the instructions you give your users and your workflows with this in mind, to figure out what you need to change.
- If you are currently using `Save` with `Update TM` at the end of your workflows to ensure that the TM is updated, consider adding a `Set Translation Status` auto action step just prior to `Save`, to set all segment translation statuses to **Reviewed**.
- If you are using TRADOS or some other tool, on import all translated segments will be set to **Pending Review**. If you consider the work your LSP does to be “Reviewed”, then put a `Set Translation Status` auto action in the workflow after the LSP to set all segments to **Reviewed**.

- Do not be concerned if you realize you have made a mistake and a project finishes but some or all of the segments were never set to **Reviewed**. Use the translation memory management page to find the appropriate entries in TM and set them to **Reviewed**. Unlike the old model, you rarely need to worry about completely losing translation memory.
- Monitor your log files for warnings. If you are using steps in your workflow that are explicitly updating translation memory, you will receive a warning. You can use these warnings to improve your process.

Chapter 3

TM Mode Implementation

For WorldServer 2011, users need to be aware of the two types of translation memory (TM) modes for several reasons, including :

- New default settings in the `tm.properties` file.
- Project settings changes that will occur in Studio when you open a project.
- Auto translation and machine translation defaults.
- Various page changes in the WorldServer user interface.

TM Mode Settings

In Studio the settings that handle TM mode mechanisms are available in the user interface. In WorldServer, you set these defaults in the `tm.properties` file. This self-documenting file is found in the WorldServer installation, in the following folder: `[WorldServer home]\tomcat\webapps\ws\WEB-INF\classes\config\tm.properties`.

To find the settings that apply to the Studio-aligned mode, search for entries that start with **`STUDIO_`**.

Note: In the Studio-aligned mode, the default settings are the same as the SDL Trados Studio defaults.

Note: Settings are incorporated in the package and applied by Studio.

Auto Substitution Settings

No replacement options are available for date and time, or number auto-localization. All information required for auto-localization is taken from the document source segment.

For measurements, different options are available which mostly control whether, and which, whitespace is inserted between the numeric value and the unit of an auto-localized measurement. The whitespace setting can be as in the document source segment, as in the memory target segment (if available), or be a fixed setting. (In other words, never insert whitespace, or always insert [some specific] whitespace character).

Project Settings Changes in Studio

When you open a WorldServer project package in Studio, several settings of the WorldServer instance which was used to create the kit will be applied to the project settings.

The project settings in Studio will be changed for the currently open project. The settings include:

- Penalties, such as the Multiple Translations Penalty, Missing Formatting Penalty, Different Formatting Penalty, and the Auto-Localization Penalty.
- The settings which control auto-substitution.
- The minimum match value which was set by the WorldServer user.
- The Fuzzy Bands which were defined by the WorldServer user (in scoping configuration).

In addition, the translation memory (TM) which is imported as part of the project has the same token recognizer settings as were used by the underlying WorldServer TM.

Auto Translation in Studio-Aligned Mode

By default, the auto translation settings in WorldServer's Studio-aligned TM mode match the defaults in SDL Trados Studio. Modify any of these with care.

New users in WorldServer 2011, or anyone who chooses the Studio-aligned TM mode, will have the following defaults, that are equivalent to SDL Studio settings:

- The recognizers are turned on.
- Auto substitution is turned on.

Machine Translation Details

For consistency between WorldServer scoping and Studio scoping reports in projects using machine translation (MT) services, the MT fuzzy score will need to be configured to be 50% (the same as Studio scoping configuration LowestScopingRange.Max) or the active scoping configuration's LowestScopingRange.Max value. This ensures that lower-scoring WorldServer TM matches do not get lost in WorldServer scoping reports (hidden by MT matches).

Note: The default MT score is typically set to 75% or 85%, depending on the MT service used.

For MT integration, use the automatic action **Apply Machine Translation** in workflows before you create Studio product packages. This pre-translates an asset using only MT, if there is not a TM match that has a score greater than the user specified minimum score for MT pre-translation. By default, this already has the desired minimum match threshold of 50% (which is also configurable).

User Interface Changes

You select different TM modes from the **Scoping Mode** radio button on the following pages:

- **Explorer** > `Asset(s)` > **Project** > **Create Project**
- **Explorer** > `Asset(s)` > **Project** > **Scope**
- **Explorer** > `Asset(s)` > **Tools** > **Export/Import** > **Export Translation Kit** > **Create Translation Kit**
- **Explorer** > `Asset(s)` > **Edit** > **In Browser Workbench** > **Select Asset version**
- **Management** > **Business Rule Linkage** > **Project Types page** > **ProjectName** > **Processing Tab**

Read-only indicators that identify the modes appear on the following pages.

- **Assignments** > **Projects** > *`ProjectName`* > **View scoping information...**
- **Assignments** > **Projects** > *`ProjectName`* > **View project information...**
- **Assignments** > **Projects** > *`ProjectName`* > **View translation progress information...**
- **Transport** > **Quotes pages**

Real-Time Studio Updates

With Studio 2014, you can see real-time updates to the project in WorldServer as you work through it in Studio. The translation memory attached to a WorldServer 10.3 package is centrally shared between WorldServer and Studio. When opening a WorldServer package, Studio users with access to the WorldServer system also have real-time access to the server-based TM attached to the package. Each update to a segment or translation memory will be immediately reflected on the WorldServer side.

Note: The following restrictions apply to the real-time updates:

- WorldServer must be in Live TM mode.
- The **Update** checkbox must be selected in Studio in the **Options** dialog box under **Translation Memory and Automated Translation** for the TM you are using.
- Changes to source segments are not supported.
- Split and merged segments will not be updated.
- Any changes to tags will not be updated.
- Offline mode is not supported.
- Segments that were originally locked will not be updated

Chapter 4

Studio Examples

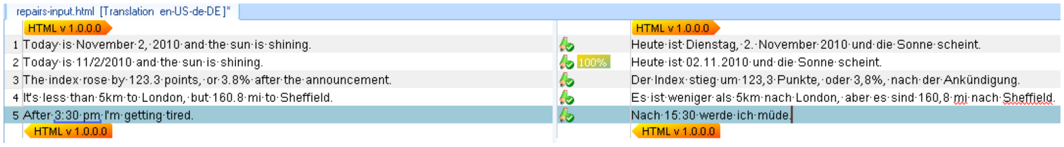
The following examples show translations in SDL Trados Studio, with notes on how the handling is affected by the translation memory mode.

Recognized Token Example

The following input document is translated in Studio, with default settings, against an empty TM:

```
<html><body>
<p>Today is November 2, 2010 and the sun is shining.</p>
<p>Today is 11/2/2010 and the sun is shining.</p>
<p>The index rose by 123.3 points, or 3.8% after the announceme
nt.</p>
<p>It’s less than 5km to London, but 160.8 mi to Sheffield.<
/p>
<p>After 3:30 pm I’m getting tired.</p>
</body></html>
```

After translation in Studio, the document looks like this:



repairs-input.html [Translation en-US-de-DE]*

	HTML v 1.0.0.0		HTML v 1.0.0.0
1	Today is November 2, 2010 and the sun is shining.		Heute ist Dienstag, 2. November 2010 und die Sonne scheint.
2	Today is 11/2/2010 and the sun is shining.	100%	Heute ist 02.11.2010 und die Sonne scheint.
3	The index rose by 123.3 points, or 3.8% after the announcement.		Der Index stieg um 123,3 Punkte, oder 3,8%, nach der Ankündigung.
4	It's less than 5km to London, but 160.8 mi to Sheffield.		Es ist weniger als 5km nach London, aber es sind 160,8 mi nach Sheffield.
5	After 3:30 pm I'm getting tired.		Nach 15:30 werde ich müde.
	HTML v 1.0.0.0		HTML v 1.0.0.0

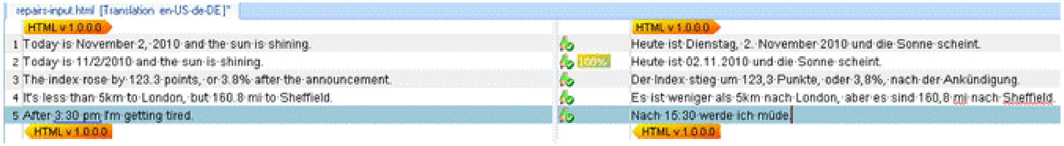
Notes:

- All placeables (indicated by a blue bracket under the placeable) have been automatically localized by Studio and selected from the segment-specific drop-down list (Ctrl-Comma).
- The second segment in fact is an exact match against the first segment. The source of the first segment uses a long date pattern, while the second uses a short date pattern, but for auto-substitution purposes, only the type (date) is compared, not the specific pattern type (long date vs. short date).
- The TM resulting from translating the above document will contain four TUs – one for segments 1 and 2 (which are "equal after auto-substitution"), and one for each of the other segments.

Now, the following document is opened in Studio. The placeables differ textually, but the other text is the same:

```
<p>Today is October 11, 2010 and the sun is shining.</p>
<p>Today is 10/11/2010 and the sun is shining.</p>
<p>The index rose by 23.45 points, or 1.3% after the announceme
nt.</p>
<p>It’s less than 4.3km to London, but 1,6000.99 mi to
Sheffield.</p>
<p>After 1:12 AM I’m getting tired.</p>
</body></html>
```

When pre-translating these documents in Studio, you get the following result, without any manual input:



repairs-input.html [Translation en-US-de-DE]*

	HTML v 1.0.0.0		HTML v 1.0.0.0
1	Today is November 2, 2010 and the sun is shining.		Heute ist Dienstag, 2. November 2010 und die Sonne scheint.
2	Today is 11/2/2010 and the sun is shining.	100%	Heute ist 02.11.2010 und die Sonne scheint.
3	The index rose by 123.3 points, or 3.8% after the announcement.		Der Index stieg um 123,3 Punkte, oder 3,8%, nach der Ankündigung.
4	It's less than 5km to London, but 160.8 mi to Sheffield.		Es ist weniger als 5km nach London, aber es sind 160,8 mi nach Sheffield.
5	After 3:30 pm I'm getting tired.		Nach 15:30 werde ich müde.
	HTML v 1.0.0.0		HTML v 1.0.0.0

Notes:

- The text of all recognized tokens (number, dates, times, measurements) differs, but all tokens are correctly inserted into the translation in their auto-localized form.
- All matches are context matches (similar to the ICE match type in WorldServer). That is, the CM match property *"survives"* repairs (which has pros and cons).
- In Studio, a 100% match which was repaired to become an exact match is not considered an original fuzzy match. That is, the match candidates are found by exact search, not by fuzzy search, since the index only stores *"placeholders"* for the token types, and does not depend on the actual textual form of the token.

- The Studio TM will only store and process one *"version"* of two translation units which are equal, apart from auto-substitutable recognized tokens. That is, even after confirming all 5 TUs in the newly translated document, the TM will only contain the 4 TUs generated by translating the original document.
- The text context hashes used to determine the CM property are independent of recognized tokens. This means that after translating the second document, the text context hashes annotated to the 4 TUs in the TM will not be changed or extended.

Markup Tags Example

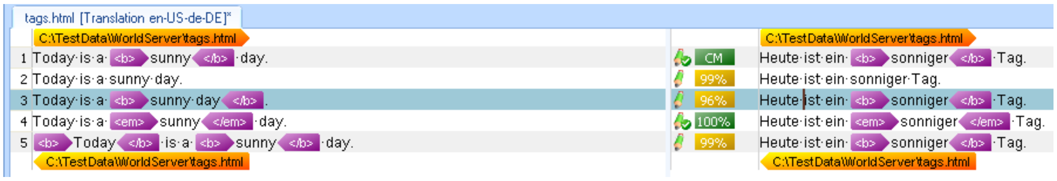
The following document is translated against an empty TM in Studio, using the tags in the translation:

```
<html><body>
<p>Today is a <b>sunny<\b> day.</p>
</body></html>
```

Then, the following document is translated against the TM produced by the previous step:

```
<html><body>
<p>Today is a <b>sunny<\b> day.</p>
<p>Today is a sunny day.</p>
<p>Today is a <b>sunny day<\b>.</p>
<p>Today is a <em>sunny day<\em>.</p>
<p><b>Today<\b> is a <b>sunny<\b> day.</p>
</body></html>
```

In Studio, this leads to the following translated document:



tags.html [Translation en-US-de-DE]*

	C:\TestData\WorldServer\tags.html		C:\TestData\WorldServer\tags.html
1	Today is a <b> sunny </b> day.	CM	Heute ist ein <b> sonniger </b> Tag.
2	Today is a sunny day.	99%	Heute ist ein sonniger Tag.
3	Today is a <b> sunny day </b> .	96%	Heute ist ein <b> sonniger </b> Tag.
4	Today is a <em> sunny </em> day.	100%	Heute ist ein <em> sonniger </em> Tag.
5	<b> Today </b> is a <b> sunny </b> day.	99%	Heute ist ein <b> sonniger </b> Tag.
	C:\TestData\WorldServer\tags.html		C:\TestData\WorldServer\tags.html

Notes:

- The first segment is a literal copy of the originally translated segment and just repeated here.
- The second segment has all memory tags deleted, triggering the missing formatting penalty.
- In the third segment, the closing tag was moved, resulting in a 96% match.
- In the fourth segment, the original boldface formatting was changed to emphasis, which still leads to a 100% match, since the actual tag contents are taken from the document and auto-substituted into the translation.
- In the fifth segment, an additional tag was introduced, leading to the "different formatting penalty" to be applied once.