



LocalizeDirect Socket API – Message Specifications

LocalizeDirect Socket API – Message Specifications	2
GetStrings Message	3
Compact Data	7
UpdateStrings Message	9
InsertStrings Message	12
StringImport Message	15
StringExport Message	18
DeleteStrings Message	22
MoveStrings Message	24
ChangeStringFolderStatus Message	26
CreateStringFolder Message	28
CreateUser Message	30
GetHistory Message	32
GetCustomFields Message	34
GetCharacters Message	36
GetStringFolders Message	37

LocalizeDirect Socket API – Message Specifications

This document describes the various messages that can be sent to and received from the LocalizeDirect server. The messages are described in this document as pseudo XML, in the sense that optional fields are denoted by [].

For more information about how to communicate with LocalizeDirect using the Socket API, please refer to the document named *LocalizeDirect Socket API.pdf*.

A note on terminology in this document:

Where used in this document, the term GUID refers to a string that is formatted as a 36 character long Globally Unique Identifier (eg. 9D3E82CB-BF66-C412-94E3-D2A77937EEF5).

GetStrings Message

This message is used to fetch strings from the server. The message has a number of arguments that can be used to specify the scope of strings to fetch. It's also possible to specify which data object fields to fetch.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="GetStrings">
    <OBJECT name="String">
      [<identifierName/>]
      [<text/>]
      [<stringId/>]
      [<stringType/>]
      [<folderPath/>]
      [<folderId/>]
      [<description/>]
      [<lastModified/>]
      [<folderName/>]
      [<requiredContent/>]
      [<audioStatus/>]
      [<textImageStatus/>]
      [<dependentFileStatus/>]
      [<custom field name/>]
    </OBJECT>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      [<compactData/>]
      [<lastModified/>]
      [<languageCode/>]
      [<folderName/>]
      [<folderId/>]
      [<recursive/>]
      [<identifierName/>]
      [<stringId/>]
      [<stringType/>]
      [<excludeOutOfDate/>]
      [<includeStatus/>]
      [<custom field name/>]
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
identifierName	(Optional) String Identifier Name (If non unique string identifiers are used in the project, then the full path must be specified).
text	(Optional) String or translation text
stringId	(Optional) String Id as a GUID
stringType	(Optional) String type
folderPath	(Optional) Folder Path
folderId	(Optional) Folder Id as a GUID
description	(Optional) Description
lastModified	(Optional) Date and time when the string was last modified.
requiredContent	(Optional) Required content specification.

	<p>Specifies the required content by a value that is a logical OR of the following (depending on the requirements for the string):</p> <p>0x1 = Audio file required 0x2 = Text Image required 0x4 = Dependent file required.</p> <p>Example: A value of 7 means audio, text and dependent file is required. A value of 1 means that an audio file is required.</p>
audioStatus	<p>(Optional) Audio status. Can be one of:</p> <p>0 = No audio file 2 = Out of date 4 = Final</p> <p>Note: This status will be returned as an attribute to the text-element.</p>
textImageStatus	<p>(Optional) Text Image status. Can be one of:</p> <p>0 = No text image file 2 = Out of date 4 = Final</p> <p>Note: This status will be returned as an attribute to the text-element.</p>
dependentFileStatus	<p>(Optional) Dependent File status. Can be one of:</p> <p>0 = No dependent file 2 = Out of date 4 = Final</p> <p>Note: This status will be returned as an attribute to the text-element.</p>
<custom field name>	<p>(Optional) Name of a custom field to be fetched. More than one custom field name can be specified in the OBJECT-element.</p>

WHERE elements	
projectName	Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.
compactData	(Optional) If set to true, then data will be returned in a compact way instead of XML formatted. <i>See description</i>

	<i>below "Compact Data".</i>
lastModified	<p>(Optional) Specifies that only strings modified after the specified date and time should be fetched. The time must be specified in the server time zone and the format must be:</p> <p>yyyy-mm-dd hh:mm:ss</p> <p>Note: The GetStrings-message will always return an element called <CurrentDateTime> that holds the server time for when the strings of this message was fetched from the server. This time can then be used as lastModified-argument to fetch strings that's been changes since last time this message was invoked.</p>
languageCode	<p>(Optional) One or many language codes of languages to fetch. The language is a 4 character string that identifies the language, for example "enUS" means American English and "frFR" means French.</p> <p>To specify more than one languageCode, use semicolon (;) as separator. For example: enUS;frFR</p> <p>Note: If this argument isn't specified as a WHERE-argument then all languages will be fetched.</p>
folderPath	<p>(Restricted Optional) Specifies that only strings in the specified folders should be fetched. The folder paths must be valid and start with the root string folder. Use a slash (/) to separate folders in the path. For example:</p> <p>Strings/Folder/Sub Folder</p> <p>Use semicolon (;) to specify more than one folder path.</p> <p>Note: <i>folderPath</i> should only be used if <i>folderId</i> isn't specified. (Don't use both <i>folderId</i> and <i>folderPath</i> as they are alternatives to each other and the one will override the other depending on the order they appear in the message).</p>
folderId	<p>(Restricted Optional) Specifies that only strings in a specific folder should be fetched. The folderId must be a valid GUID. If more than one folder needs to be specified, then use a semicolon (;) as separator.</p> <p>Note: <i>folderId</i> should only be used if <i>folderPath</i> isn't specified. (Don't use both <i>folderId</i> and <i>folderPath</i> as they are alternatives to each other and the one will override the other depending on the order they appear in the message).</p>
recursive	<p>(Optional) This value can either be set to "true" or "false" (default is "false"). The value specifies whether or not strings should be fetched recursively under the folders specified either using the <folderId> or <folderPath> element.</p>

	(Recursive in this case means that strings in sub-folders also will be fetched.)
identifierName	(Optional) Specifies one or many strings to be fetched using identifier name. Use semicolon (;) to specify more than one identifier name.
stringId	(Optional) Specifies one or many strings to be fetched using string id (GUID). Use semicolon (;) to specify more than one string id.
stringType	(Optional) String Type.
excludeOutOfDate	(Optional) This value can be set to either “true” or “false”. If it’s set to “true”, then all translations that are out-of-date will be excluded from the result.
includeStatus	<p>(Optional) This value can be set to either “true” or “false”. If it’s set to “true”, then the translation status for each translation will be returned. If non-compact data sets is used, then the status will be returned as an attribute to the <text> element. If compact data sets are used, then an additional status-column will be added for each language.</p> <p>The status can be one of the following values:</p> <p>0 = Source Language 1 = Not translated 2 = Out-of-date 3 = Translated (up-to-date)</p>
<custom field name>	(Optional) Custom field and value to filter on.

RESULTSET structure

The elements in the RESULTSET-structure may vary depending on the fields specified in the OBJECT-element of this message.

```

<RESULTSET>
  <DATASETS>
    <Strings>
      <currentDateTime>2011-08-22 13:25:30</currentDateTime>
      <UTC_offset>1</UTC_offset>
      <String>
        <identifierName>STR_APPLY</identifierName>
        <text language="esES">Aplicar</text>
        <text language="enUS">Apply</text>
        <text language="frFR">Appliquer</text>
        <text language="deDE">Anwenden</text>
        <text language="itIT">Applica</text>
        [Additional fields specified in OBJECT element]
      </String>
      ...
    </Strings>
  </DATASETS>

```

```
<RESULT/>
</RESULTSET>
```

Strings element	
currentDateTime	The server date time for when the strings were fetched from the server.
UTC_offset	The server UTC (Coordinated Universal Time) offset (in hours).

String elements (default)	
identifierName	String Identifier Name
text	String text for the language specified by the <i>language</i> -attribute.

The result node will contain MESSAGE-elements if something went wrong.

Compact Data

If the WHERE-argument `<compactData>` is set to “true”, the returning string data will be returned in a more compact way compared to when returned as XML formatted data. Instead of having String xml elements with child elements for each column, all of the data will be returned in one single xml element called `<data>`. Each of the returning strings will be separated with pairs of 0xAF (Unicode characters: ¯) and the data of each string will be separated with pairs of 0xAC (Unicode characters: ¨).

An additional XML element called `<metadata>` will be returned together with the `<data>` element. The `<metadata>` element will describe the name and the order of the data returned in the `<data>` element.

This is an example of what the data may look like when using compact data:

```
<RESULTSET>
<DATASETS>
  <Strings>
    <metadata>identifierName¨¨text_enEN¨¨text_deDE¨¨</metadata>
    <data>STR_APPLY¨¨Apply¨¨Anwenden¨¨STR_SEARCH¨¨Search¨¨Suchen¨¨ </data>
    ...
  </Strings>
</DATASETS>
</RESULTSET>
```

In the example above, two strings are returned. Note that the metadata element will only describe one single “row of data”, as all returning strings will be returned this way (eg. the same columns and the same order of the columns).

Note that the *language*-attribute that is used when string data is returned as XML formatted now instead is described in the metadata as `text_<language code><contry code>` (eg. `text_enUS` for American English and `text_deDE` for German).

UpdateStrings Message

This message is used to update the text or data of one or many strings.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="UpdateStrings">
    <OBJECT name="String">
      [<identifierName/>]
      [<sourceLanguageText/>]
      [<text_<i>languageCode</i>/>]
      [<description/>]
      [<custom field name/>]
      [<order>]
      [<stringType>]
      [<retranslate>]
      [<requiredContent/>]
      [<translationStatus_<i>languageCode</i>/>]
      [<audioStatus_<i>languageCode</i>/>]
      [<textImageStatus_<i>languageCode</i>/>]
      [<dependentFileStatus_<i>languageCode</i>/>]
    </OBJECT>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      [<identifierName/>]
      [<stringId/>]
    </WHERE>
  </TASK>

  ...
  [<Additional UpdateStrings-TASK for another string>]
</EXECUTION>
```

Example of how to update a string:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="UpdateStrings">
    <OBJECT name="String">
      <sourceLanguageText>testing</sourceLanguageText>
    </OBJECT>
    <WHERE>
      <projectId>31FB05E1-F3B0-7AEF-0508-46162DFE987D</projectId>
      <identifierName>Strings_1</identifierName>
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
identifierName	(Optional) New String Identifier Name (If non unique string identifiers are used in the project, then the full path must be specified).
sourceLanguageText	(Optional) New Source Language Text.
text_<i>languageCode</i>>	(Optional) New String translation text to be updated. The <i>languageCode</i> section of the name should be replaced with the code of the language of the translation, e.g. text_frFR or

	text_esES.
description	(Optional) Description
<custom field name>	(Optional) New Custom field value to be updated.
order	(Optional) New String order (only applies to Dialogue Strings)
stringType	(Optional) String Type
retranslate	(Optional) If the source language text is changed and it has already been translated, then this parameter can be used to indicate whether or not translations need to be retranslated. The value can be either <i>true</i> or <i>false</i> . Not specifying this parameter (not including it in the message) is equal to setting it to <i>false</i> .
requiredContent	<p>(Optional) Required content specification. Specify the required by a value that is a logical OR of the following (depending on the requirements for the string):</p> <p>0x1 = Audio file required 0x2 = Text Image required 0x4 = Dependent file required.</p> <p>Example: A value of 7 means audio, text and dependent file is required. A value of 1 means that an audio file is required.</p>
translationStatus_<languageCode>	<p>(Optional) Updates the status of the translation with the specified language code. The status may be set to one of the following.</p> <p>2 = Out-of-date 3 = Translated (up-to-date)</p> <p>Note: Status 0 (meaning Source Language) and 1 (meaning no translation) may not be set using this message.</p>
audioStatus_<languageCode>	<p>(Optional) Audio status. Can be one of:</p> <p>0 = No audio file 2 = Out of date 4 = Final</p>
textImageStatus_<languageCode>	<p>(Optional) Text Image status. Can be one of:</p> <p>0 = No text image file 2 = Out of date 4 = Final</p>
dependentFileStatus_<languageCode>	<p>(Optional) Dependent File status. Can be one of:</p> <p>0 = No dependent file</p>

	2 = Out of date 4 = Final
--	------------------------------

WHERE elements	
projectName	(Restricted Optional) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Restricted Optional) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.
identifierName	(Restricted Optional) String Identifier Name (If non unique string identifiers are used in the project, then the full path must be specified). Using a semicolon (;) separator, more than one identifier name may be specified in this field.
stringId	(Restricted Optional) Unique String Id (GUID). Must be specified unless <i>identifierName</i> is specified. Using a semicolon (;) separator, more than one string id may be specified in this field.

RESULTSET structure

If successfully updated, then the result set will consist of a RESULT-element with no elements, else there will be a MESSAGE-element under the RESULT-element.

Successful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT />
</RESULTSET>
```

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="1" type="Error">
      Found no String matching the specified projectId and
      identifierName!
    </MESSAGE>
  </RESULT>
</RESULTSET>
```

InsertStrings Message

This message is used to insert new strings into Localize Direct.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="InsertStrings">
    <OBJECT name="String">
      <identifierName/>
      [<sourceLanguageText/>]
      [<text_<i>languageCode</i>/>]
      [<description/>]
      [<custom field name/>]
      [<order/>]
      [<stringType/>]
      [<requiredContent/>]
      [<audioStatus_<i>languageCode</i>/>]
      [<textImageStatus_<i>languageCode</i>/>]
      [<dependentFileStatus_<i>languageCode</i>/>]
    </OBJECT>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      [<folderName/>]
      [<folderId/>]
    </WHERE>
  </TASK>

  ...
  [<Additional InsertStrings-TASK for another string>]
</EXECUTION>
```

Example of how to insert a source language string text:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="InsertStrings">
    <OBJECT name="String">
      <identifierName>Strings/Test/str_test</identifierName>
      <sourceLanguageText>testing</sourceLanguageText>
    </OBJECT>
    <WHERE>
      <projectName>Test Project</projectName >
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
identifierName	String Identifier Name (If non unique string identifiers are used in the project, then the full path must be specified).
sourceLanguageText	(Optional) Source Language Text.
text_<i>languageCode</i>	(Optional) String translation text to be inserted. The <i>languageCode</i> section of the name should be replaced with the code of the language of the translation, e.g. text_frFR or text_esES.

description	(Optional) Description
<custom field name>	(Optional) Custom field to be inserted.
order	(Optional) String order (only applies to Dialogue Strings)
stringType	(Optional) String Type
requiredContent	(Optional) Required content specification. Specify the required by a value that is a logical OR of the following (depending on the requirements for the string): 0x1 = Audio file required 0x2 = Text Image required 0x4 = Dependent file required. Example: A value of 7 means audio, text and dependent file is required. A value of 1 means that an audio file is required.
audioStatus_<languageCode>	(Optional) Audio status. Can be one of: 0 = No audio file 2 = Out of date 4 = Final
textImageStatus_<languageCode>	(Optional) Text Image status. Can be one of: 0 = No text image file 2 = Out of date 4 = Final
dependentFileStatus_<languageCode>	(Optional) Dependent File status. Can be one of: 0 = No dependent file 2 = Out of date 4 = Final

WHERE elements	
projectName	(Restricted Optional) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Restricted Optional) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.
folderName	(Optional) Specifies the name of the folder under which the strings should be inserted. If a path was specified in the <i>identifierName</i> (object-element) then there's no need to specify the <i>folderId</i> element in the message.
folderId	(Optional) Specifies the unique folder id (GUID) under which the strings should be inserted. If a path was specified in the <i>identifierName</i> (object-element) then there's no need to specify the <i>folderId</i> element in the message.

RESULTSET structure

If successfully inserted, then the result set will consist of a RESULT-element with no elements, else there will be a MESSAGE-element under the RESULT-element.

Successful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT />
</RESULTSET>
```

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="1" type="Error">
      Failed to insert the string message
    </MESSAGE>
  </RESULT>
</RESULTSET>
```

StringImport Message

As an alternative to the InsertStrings and UpdateStrings messages, the StingImport message can be used. This message is much more efficient when it comes to importing many strings at once. The message operates very much in the same way as the Import Strings feature in the client.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="StringImport">
    <OBJECT name="String">
      <importFields/>
      <fieldData/>
    </OBJECT>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      <stringMergeOption/>
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
importFields	<p>(Required) A semicolon (;) separated list of the fields to be imported. The data of these field will then be specified for each string in the fieldData-element where separators are being used to separate the stings and the values of the strings.</p> <p>When importing strings there are two fields that always is required: <i>folderPath</i> and <i>identifierName</i>. All other fields are optional.</p> <p>Custom fields are simply just specified with the name of the custom field and translation texts can be specified as <i>text_<language code></i>, for example <i>text_frFR</i> for a French translation.</p>
fieldData	<p>(Required) This is the data for all strings that will be imported.</p> <p>The data for each field is separated with a pair of Unicode 0xAC characters (↯↯) and each string is separated with a pair of Unicode 0xAF characters (¯¯). The reason for why these special characters are being used, is to allow for more traditional separators like semicolon and comma to be used as field values (in for example the string texts).</p>


```
<RESULTSET>
  <DATASETS>
    <DATASET datatype="dataSet">
      <numberOfImportedStings>3</numberOfImportedStings>
      <numberOfFailedStings>0</numberOfFailedStings>
    </DATASET>
  </DATASETS>
</RESULTSET>
```

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="7" type="Error">
      A string with identifier STR_123 already exists!
    </MESSAGE>
  </RESULT>
</RESULTSET>
```


StringExport Message

As an alternative to the GetStrings messages, the StingExport message can be used. This message works in a similar way as when using the CSV exporter on the client side. It's possible to export all string fields and it's possible to chose the response type.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="StringExport">
    <OBJECT name="String">
      <exportFields/>
    </OBJECT>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      [<folderPaths/>]
      [<identifiers/>]
      <responseType/>
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
exportFields	<p>(Required) A semicolon (;) separated list of the fields to be exported.</p> <p>Custom fields are simply just specified with the name of the custom field and translation texts can be specified as <i>text_<language code></i>, for example <i>text_frFR</i> for a French translation.</p> <p>For a full list of fields that can be exported see the list below.</p>

WHERE elements	
responseType	<p>(Required) Specifies the format of the response data.</p> <p>The value can be one of the following:</p> <p>xml = XML formatted data. csv = CSV formatted data. (Comma Separated Values) ldc = Localize Direct Compact data format.</p> <p>Note: Using the ldc-format, quotes and other special characters will not be treated in a special way. Instead unique separators will be used. The data for each field is separated with a pair of Unicode 0xAC characters (↵↵) and each string is separated with a pair of Unicode 0xAF characters (¯¯).</p>
projectName	<p>(Required see Text) Project Name. Must be specified unless <i>projectId</i> is specified.</p>

projectId	(Required see Text) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.
folderPaths	<p>(Required see Text) Specifies the folders to be exported. The folder paths must be valid and start with the root string folder. Use a slash ('/') to separate folders in the path. For example:</p> <p>Strings/Folder/Sub Folder</p> <p>Use semicolon (;) to specify more than one folder path.</p> <p>Note: <i>folderPaths</i> should always be specified unless <i>identifierName</i> is used instead to only export specific strings.</p>
identifierName	<p>(Optional) String Identifier Name of strings to be exported. (If non unique string identifiers are used in the project, then the full path must be specified).</p> <p>This option can be used if only specific strings should be exported.</p> <p>Using a semicolon (;) separator, more than one identifier name may be specified in this field.</p>

List of Fields that can be exported

Export Field Name	Description (Meaning and response)
stringId	String Globally Unique Identifier (GUID)
path	String Path
identifierName	String Identifier Name
<i>Custom Field Name</i>	Custom field name of any custom field that should be exported
<i>text_<language code></i>	Translation text. This name should be specified as <i>text_<language code></i> , for example <i>text_frFR</i> for a French translation.
<i>status_<language code></i>	<p>Translation status. This name should be specified as <i>status_<language code></i>, for example <i>status_frFR</i> for a French translation.</p> <p>The response value is a numeric value with the following meaning:</p> <p>1 = Not translated (Red in client) 2 = Out-of-date (Yellow in client) 3 = Translated (Green in client)</p>
stringType	A numeric value indicating the string type to which this string belongs.
stringTypeName	A string value with the name of the String Type to which this string belongs.

description	String description
childOrder	String child order under it's parent (only automatically set by LD if Dialogue Folders are being used).
maxPixelWidth	Max Pixels Dimensions
maxTextLength	Max Text Dimensions
translationPriority	Numeric value indicating the string translation priority.
requiredContent	Indicates if this String has required content.
tokenRule	Token rule defined for the String.
hasAttachment	Boolean value that indicates if the string has a content attachment.
hasComments	Boolean value that indicates if the string has comments.
lastModified	Date when source string data was last modified (not translations)

List of Character specific Fields that can be exported

The list below is names from character objects that also can be exported. These fields are actually not stored on each string, but can still be accessed this way.

Export Field Name	Description (Meaning and response)
dialogueCharacterId	GUID of character object associated with string.
dialogueCharacterName	Character Name
dialogueCharacterType	Character Type
characterDescription	Character Description
characterAccent	Character Accent/ethnicity
characterNotes	Character Notes
characterDefine	Character Define
characterMisc	Character Misc
characterCustom	Character Custom
characterFinalActor	Character Final Actor
characterCharacteristics	Character Characteristics
characterProcess	Character Process
characterActorMixer	Character Actor Mixer
characterBrand	Character Brand
characterOfficeVoice	Character Office Voice
characterOfficeActor	Character Office Actor
characterStuntActor	Character Stunt Actor
characterRobotVoice	Character Robot Voice
characterOfficeVoiceSettings	Character Office Voice Settings
characterOfficeActorSettings	Character Office Actor Settings
characterFinalActorSettings	Character Final Actor Settings
characterStuntActorSettings	Character Stunt Actor Settings

Example of a StringExport message:

This is an example of a string export message::

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="StringExport">
    <OBJECT name="String">
<exportFields>path;identifierName;sourceLanguageText;status_frFR;text_frFR;
MyCustom;</exportFields>
    </OBJECT>
    <WHERE>
      <projectName>My Game</projectName>
      <folderPaths>Strings/Test/A;/Strings/Test2/;</folderPaths>
      <identifiers></identifiers>
      <responseType>csv</responseType>
    </WHERE>
  </TASK>
</EXECUTION>
```

Response

CSV

The following is an example of response in XML:

```
"Strings/Example/","STRING_1","Text 1","2","French Text 1"
"Strings/Example/","STRING_2","Text 2","3","French Text 2"
"Strings/Example/","STRING_3","Text 3","3","French Text 3"
```

LDC

The following is an example of response in XML:

```
Strings/Example/--STRING_1--Text 1--2--French Text
1--`Strings/Example/--STRING_2--Text 2--3--French Text
2--`Strings/Example/--STRING_3--Text 3--3--French Text 3--`
```

XML

The following is an example of response in XML:

```
<RESULTSET>
  <DATASETS>
    <currentDateTime>2014-03-14 10:49:22</currentDateTime>
    <UTC_offset>1</UTC_offset>
    <![CDATA[Strings/Example/--STRING_1--Text 1--2--French Text
1--`Strings/Example/--STRING_2--Text 2--3--French Text
2--`Strings/Example/--STRING_3--Text 3--3--French Text 3--`]]>
  </DATASETS>
</RESULTSET>
```

DeleteStrings Message

This message is used to delete strings from Localize Direct.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="DeleteStrings">
    <OBJECT name="String"/>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      [<identifierName/>]
      [<stringId/>]
    </WHERE>
  </TASK>

  ...
  [<Additional DeleteStrings-TASK for another string>]
</EXECUTION>
```

Example of how to delete strings:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="DeleteStrings">
    <OBJECT name="String"/>
    <WHERE>
      <projectName>Test Project</projectName >
      <identifierName>str_test1;str_test2</identifierName>
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements (Non)

WHERE elements	
projectName	(Restricted Optional) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Restricted Optional) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.
identifierName	(Restricted Optional) String Identifier Name (If non unique string identifiers are used in the project, then the full path must be specified). Must be specified unless <i>stringId</i> is specified. Using a semicolon (;) separator, more than one identifier name may be specified in this field.
stringId	(Restricted Optional) Unique String Id (GUID). Must be specified unless <i>identifierName</i> is specified. Using a semicolon (;) separator, more than one string id may be specified in this field.

RESULTSET structure

If successfully deleted, then the result set will consist of a RESULT-element with no elements, else there will be a MESSAGE-element under the RESULT-element.

Successful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT />
</RESULTSET>
```

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="1" type="Error">
      Failed to delete string message
    </MESSAGE>
  </RESULT>
</RESULTSET>
```

MoveStrings Message

This message is used to move strings from one folder to another.

Note: To move or change the order of a Dialogue String, use the updateStrings-message instead.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="MoveStrings">
    <OBJECT name="String">
      [<folderName/>]
      [<folderId/>]
    </OBJECT>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      [<identifierName/>]
      [<stringId/>]
    </WHERE>
  </TASK>

  ...
  [<Additional MoveStrings-TASK for another string>]
</EXECUTION>
```

Example of how to move strings:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="MoveStrings">
    <OBJECT name="String">
      <folderName>Strings/Test/</folderName>
    </OBJECT>
    <WHERE>
      <projectName>Test Project</projectName >
      <identifierName>str_test1;str_test2</identifierName>
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
folderName	(Restricted Optional) Path to the folder into which the string will be moved. Must be specified unless <i>folderId</i> is specified.
folderId	(Restricted Optional) Unique folder id (GUID) of the folder into which the string will be moved. Must be specified unless <i>folderName</i> is specified.

WHERE elements	
projectName	(Restricted Optional) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Restricted Optional) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.

identifierName	<p>(Restricted Optional) String Identifier Name (If non unique string identifiers are used in the project, then the full path must be specified). Must be specified unless <i>stringId</i> is specified.</p> <p>Using a semicolon (;) separator, more than one identifier name may be specified in this field.</p>
stringId	<p>(Restricted Optional) Unique String Id (GUID). Must be specified unless <i>identifierName</i> is specified.</p> <p>Using a semicolon (;) separator, more than one string id may be specified in this field.</p>

RESULTSET structure

If successfully moved, then the result set will consist of a RESULT-element with no elements, else there will be a MESSAGE-element under the RESULT-element.

Successful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT />
</RESULTSET>
```

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="1" type="Error">
      Failed to move string message
    </MESSAGE>
  </RESULT>
</RESULTSET>
```


ChangeStringFolderStatus Message

This message can be used to change the folder status of String Folders. The states that can be set is whether or not strings in the folder is ready for translation and whether or not the source language text is locked. The message can change the state of several folders at a time and also set the states recursively.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="ChangeStringFolderStatus">
    <OBJECT name="Folder">
      [<ReadyForTranslation/>]
      [<SourceLanguageLocked/>]
      [<applyOnChildFolders/>]
    </OBJECT>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      [<folderName/>]
      [<folderId/>]
    </WHERE>
  </TASK>
</EXECUTION>
```

Example of how to use the message:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="ChangeStringFolderStatus">
    <OBJECT name="Folder">
      <ReadyForTranslation>true</ReadyForTranslation>
      <applyOnChildFolders>true</applyOnChildFolders>
    </OBJECT>
    <WHERE>
      <projectName>Test Project</projectName>
      <folderName>Strings/Folder1;Strings/Folder2</folderName>
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
ReadyForTranslation	(Optional) <i>true</i> or <i>false</i> depending on if strings in the specified folders should be ready for translation.
SourceLanguageLocked	(Optional) <i>true</i> or <i>false</i> depending on if strings in the specified folders should have the source language text locked.
applyOnChildFolders	(Optional)) If set to <i>true</i> then the folder states will also be set on all child folders. Default is false (if this element isn't specified).

WHERE elements	
projectName	(Restricted Optional) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Restricted Optional) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.
folderName	(Restricted Optional) Path to the folder(s) to be affected. Must

	be specified unless <i>folderId</i> is specified.
folderId	(Restricted Optional) Unique folder id (GUID) of the folder to be affected. Must be specified unless <i>folderName</i> is specified.
applyOnChildFolders	(Optional) If set to <i>true</i> then the folder states will also be set on all child folders. Default is false (if this element isn't specified).

RESULTSET structure

If successfully updated, then the result set will consist of a RESULT-element with no elements, else there will be a MESSAGE-element under the RESULT-element.

Successful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT />
</RESULTSET>
```

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="1" type="Error">
      Missing folderIDs or folder paths!
    </MESSAGE>
  </RESULT>
</RESULTSET>
```

CreateStringFolder Message

Use this message to create new string folders.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="createStringFolder">
    <OBJECT name="String">
      <folderName/>
      [<description/>]
      [<folderSubType/>]
      [<folderStatus/>]
    </OBJECT>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
      [<parentFolderName/>]
      [<parentFolderId/>]
    </WHERE>
  </TASK>

  ...
  [<Additional createStringFolder-TASK>]
</EXECUTION>
```

Example of how to create a new string folder:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="createStringFolder">
    <OBJECT name="String">
      <folderName>New Folder</folderName>
      <description>A description</description>
    </OBJECT>
    <WHERE>
      <projectName>Test Project</projectName>
      <parentFolderName>Strings/Folder</parentFolderName>
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
folderName	Folder name. Must be unique under its parent folder.
description	(Optional) Description
folderSubType	(Optional) Specifies whether the folder is a string or dialogue folder. If not specified, then this will be a string folder. 0 = String Folder 1 = Dialogue Folder
folderStatus	(Optional) Specifies the default status of the folder. Use the following bits to specify a value for the folder status (default is 1): 0x1 = Strings not ready for translation.

0x2 = Source language text locked.

WHERE elements	
projectName	(Restricted Optional) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Restricted Optional) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.
parentFolderName	(Restricted Optional) Specifies the name and path of the folder under which the new string folder should be created. Must be specified unless <i>folderId</i> is specified.
parentFolderId	(Restricted Optional) Specifies the unique folder id (GUID) under which the new string folder should be created. Must be specified unless <i>parentFolderName</i> is specified.

RESULTSET structure

If successfully inserted, then the result set will return the folderId of the created folder (see example below), else there will be a MESSAGE-element under the RESULT-element.

Successful Result Example:

```
<RESULTSET>
  <DATASETS>
    <DATASET datatype="Folder">
      <folderId>98903985-8954-10D3-7956-9AC31A27DCE1</folderId>
    </DATASET>
  </DATASETS>
  <RESULT />
</RESULTSET>
```

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="0" type="Error">
      Reason for the failure message.
    </MESSAGE>
  </RESULT>
</RESULTSET>
```

CreateUser Message

Use this message to create new user.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="CreateUser">
    <OBJECT name="User">
      <userName/>
      <password/>
      <fullName/>
      [<memberOfGroup/>]
      [<memberOfProject/>]
      [<emailAddress>]
      [<description/>]
    </OBJECT>
    <WHERE>
      [<parentFolderName/>]
      [<parentFolderId/>]
    </WHERE>
  </TASK>

  ...
  [<Additional createStringFolder-TASK>]
</EXECUTION>
```

Example of how to create a new user folder:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="CreateUser">
    <OBJECT name="User">
      <userName>John</userName>
      <password>123</password>
      <fullName>John Smith</fullName>
      <memberOfGroup>Test Project Users</memberOfGroup>
      <memberOfProject>Test Project</memberOfProject>
    </OBJECT>
    <WHERE>
      <parentFolderName>Domain Users</parentFolderName>
    </WHERE>
  </TASK>
</EXECUTION>
```

OBJECT elements	
userName	User name. Must be unique under its parent folder.
password	User password
fullName	Full user name
memberOfGroup	(Optional) Name of one or many groups that this user is member of. Group names are separated with a semicolon (;). Note: Make sure the group names are unique on the server.
memberOfProject	(Optional) Name of projects that this user is member of. Project names are separated with a

	semicolon (;). Note: Make sure the project names are unique on the server.
emailAddress	(Optional) User Email Address
description	(Optional) User description.

WHERE elements	
parentFolderName	(Restricted Optional) Specifies the name of the folder under which the new user should be created. This is just the name of the folder NOT a path (as user folders exists outside of projects). Make sure that the name of the user folder is unique before using this parameter. Must be specified unless <i>folderId</i> is specified.
parentFolderId	(Restricted Optional) Specifies the unique folder id (GUID) under which the new user folder should be created. Must be specified unless <i>parentFolderName</i> is specified.

RESULTSET structure

If successfully inserted, then the result set will return the `userId` of the created user (see example below), else there will be a MESSAGE-element under the RESULT-element.

Successful Result Example:

```
<RESULTSET>
  <DATASETS>
    <DATASET datatype="User">
      <userId>98903982-8957-10D9-7954-9AC31A37DCE4</userId>
    </DATASET>
  </DATASETS>
  <RESULT />
</RESULTSET>
```

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="0" type="Error">
      Reason for the failure message.
    </MESSAGE>
  </RESULT>
</RESULTSET>
```

GetHistory Message

Use this message to get history data for a specific item.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="GetHistory">
    <OBJECT name="History"/>
    <WHERE>
      <dataId/>
      [<limit>]
    </WHERE>
  </TASK>

  ...
  [<Additional GetHistory-TASK>]
</EXECUTION>
```

Example of how to get item history:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="GetHistory">
    <OBJECT name="History"/>
    <WHERE>
      <dataId>E20F4D7D-021C-7E00-1DFC-E38092389F95</dataId>
    </WHERE>
  </TASK>
</EXECUTION>
```

WHERE elements	
dataId	Data Id (GUID) of the item to get history for. The data Id can be a string id, user id, group id, etc.
limit	Maximum number of rows (returning datasets) that this call should return. (E.g. if set to 10, then only the 10 latest history rows will be returned). If not specified then all rows will be returned.

RESULTSET structure

Successful Result Example:

```
<RESULTSET>
  <DATASETS>
    <DATASET datatype="History">
      <historyId>222860</historyId>
      <dataId>E20F4D7D-021C-7E00-1DFC-E38092389F95</dataId>
      <tag />
      <userId>E30F5D7D-011C-8E00-1DFC-E38092389F92</userId>
      <userName>Test User</userName>
      <countryCode />
      <eventText>Properties changed.</eventText>
      <eventType>3</eventType>
      <eventTime>2011-10-28 13:29:00.0</eventTime>
    </DATASET>
  </DATASETS>
</RESULTSET />
```

</RESULTSET>

Dataset element description

DATASET element	
historyId	Unique id
dataId	GUID of the data that this history row was generated for (for example a String GUID)
tag	Used by data objects that have additional history information (The tag is never used by String data objects).
userId	GUID of the user responsible for this action/task.
userName	Name of the user responsible for this action/task.
countryCode	If this is a history row for a string or a translation then the two character country code is stored here.
eventText	Text that describes the event.
eventType	Type of event. The event type can be one of the following: 1 = Data added 2 = Data deleted 3 = Data changed
eventTime	Time when this event occurred.

Unsuccessful Result Example:

```
<RESULTSET>
  <DATASETS />
  <RESULT>
    <MESSAGE id="0" type="Error">
      Reason for the failure message.
    </MESSAGE>
  </RESULT>
</RESULTSET>
```


GetCustomFields Message

This message will return the names and id's of the custom fields for a specific project.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="GetCustomFields">
    <OBJECT name="String"/>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
    </WHERE>
  </TASK>
</EXECUTION>
```

WHERE elements	
projectName	(Optional see Text) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Optional see Text) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.

Example of how use this message:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="GetCustomFields">
    <OBJECT name="String"/>
    <WHERE>
      <projectName>MyGame</projectName>
    </WHERE>
  </TASK>
</EXECUTION>
```

RESULTSET structure

This message will return the names and id's of the custom fields as semicolon separated lists as shown in the example below.

Successful Result Example:

```
<RESULTSET>
  <DATASETS>
    <DATASET datatype="dataSet">

<customFields>MyCustom1;MyCustom2;Test1;Test2;A_Field;New_Field;</customFields>
      <customIds>84C6CDDA-901C-F59F-DCB0-1D350A8205DB;88FE8FEB-54F8-5529-F2C0-6238CA3ED9AC;2A41B636-E36D-393C-E9B0-12F245F5EB7E;E4DEC38D-F42C-4A3F-CDEC-1EA3516FE267;18875B87-0204-2BE7-3FE1-5FC31699562E;689E2B24-A04A-5E68-83DE-E9D71A59D0F8;</customIds>
    </DATASET>
  </DATASETS> <RESULT />
</RESULTSET>
```

GetCharacters Message

This message will return all dialogue characters for a specific project.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="GetCharacters">
    <OBJECT name="DialogueCharacter"/>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
    </WHERE>
  </TASK>
</EXECUTION>
```

WHERE elements	
projectName	(Optional see Text) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Optional see Text) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.

Example of how use this message:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="GetCharacters">
    <OBJECT name="DialogueCharacter"/>
    <WHERE>
      <projectName>My Game</projectName>
    </WHERE>
  </TASK>
</EXECUTION>
```

RESULTSET structure

This message will return the all dialogue characters of the specified project, one dataset per character.

GetStringFolders Message

This message will return all String Folders in a project. Then folderId's and parentFolderId's shows the relationships between the folders (parent and children). The message will return all folders in a project, both String and Dialogue folders. To determine if a folder is a string folder or a dialogue folder, check the field named "childrenDataTypeId", 0 = String and 1 = Dialogue.

EXECUTION structure

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name=" GetStringFolders">
    <OBJECT name="Folder"/>
    <WHERE>
      [<projectName/>]
      [<projectId/>]
    </WHERE>
  </TASK>
</EXECUTION>
```

WHERE elements	
projectName	(Optional see Text) Project Name. Must be specified unless <i>projectId</i> is specified.
projectId	(Optional see Text) Unique Project Id (GUID). Must be specified unless <i>projectName</i> is specified.

Example of how use this message:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXECUTION>
  <TASK name="GetStringFolders">
    <OBJECT name="Folder"/>
    <WHERE>
      <projectName>My Game</projectName>
    </WHERE>
  </TASK>
</EXECUTION>
```

RESULTSET structure

This message will return the all string folders of the specified project, one dataset per character.