Constraints to add in the Extension Mechanism section

* When using identifiers, modules and extension MUST use either an attribute named id or the attribute xml:id to specify them.
* Modules and extensions identifiers MUST be unique within the most immediate <file>, <group> or <unit> enclosing element.
* Identifiers used in modules and extensions MUST be of type xs:NMTOKEN or compatible with xs:NMTOKEN (e.g. xs:NAME or xs:ID are compatible).

Also to do:

* Remove the prefix declaration for any modules except Matches, Glossary and Resource Data, as the others (as far as I can tell) do not have elements with IDs, and therefore can’t be accessed using fragment identification.
* Add a constraint to the id definition for Resource Data so the uniqueness of the id values is specified as unique within the proper scope as per the general module/extension constraint listed in the Extension Mechanism section (see above).
* Change the text “urn:oasis:names:tc:xliff:resourceData:2.0” in the Resource Data section to “urn:oasis:names:tc:xliff:resourcedata:2.0” (all lowercase) as it seems to be the namespace URI used elsewhere, including in the schemas.
* Start a registry file with the three prefixes for the three modules that have IDs: mtc, gls and res. I would suggest to use a properties file for the registry document: it’s a format that is well known, documented and easy to process.

# 3 Fragment Identification

Because XLIFF Documents do not follow the usual behavior of XML documents when it comes to element identifiers, this specification defines how Agents MUST interpret the fragment identifiers in IRIs pointing to XLIFF Documents.

Note that some identifiers may change during the localization process. For example <data> elements may be re-grouped or not depending on how tools treat identical original data.

Constraints:

* A fragment identifier MUST matches the following format:

<scopeSeparator> ::= "/"
<prefixSeparator> ::= "="
<prefix> ::= NMTOKEN
<id> ::= NMTOKEN
<selector> ::= [<prefix> <prefixSeparator>] <id>
<expression> ::= "#" ["/"] <selector> {<scopeSeparator> <selector>}

* There MUST NOT be two identical prefixes in the expression.
* When used, the following selectors MUST be declared in this order: file selector, group selector and unit selector.
* The selectors for modules or extensions, note, segment or ignorable or source inline elements, target inline elements and data have the following constraints:
	+ Only one of them MUST be used in the expression
	+ The one used MUST be the last selector of the expression

### Selectors for Core Elements

* The prefix 'f' indicates a <file> id and the value of that id is unique within the document.
* The prefix 'g' indicates a <group> id and the value of that id is unique within the enclosing <file> element.
* The prefix 'u' indicates a <unit> id and the value of that id is unique within the enclosing <file> element.
* The prefix 'n' indicates a <note> id and the value of that id is unique within its immediate enclosing <file>, <group> or <unit> element.
* The prefix 'd' indicates a <data> id and the value of that id is unique within its enclosing parent element.
* The prefix 't' indicates an id for an inline element in the <target> element and the value of that id is unique within the enclosing <unit> element (with the usual caveat for source/target).
* No prefix indicates an id for a <segment> or an <ignorable> or an inline element in the <source> element and the value of that id is unique within the enclosing <unit> element (with the usual caveat for source/target).

### Selectors for Modules and Extensions

A selector for a module or an extension uses a registered prefix and the value of that id is unique within the immediate enclosing <file>, <group> or <unit> element.

Constraints

* The prefix of a modules or an extension MUST be an NMTOKEN longer than 1 character and MUST be defined in the modules or extensions specification.
* The prefix of a modules or an extension MUST be registered with the XLIFF TC (See details here: <link to procedure to register a prefix>)
* A given module or extension namespace URI MUST be associated with a single prefix.
* A prefix MAY be associated with more than one namespace URI (to allows for example different versions of a given module or extension to use the same prefix).

### Relative References

Fragment identifiers that do not start with a character ‘/’ are relative to their location in the document, or to the document being processed.

Any unit, group or file selector missing to resolve the relative reference is obtained from the immediate enclosing unit, group or file elements.

### Prefix Registration for Modules and Extension

The prefixes used in the selectors for modules and extensions are registered at the OASIS XLIFF TC.

TODO: procedure to register a prefix.