Spectrum Mill BI 7.09
Creating Custom Modifications and Search Modes

A Guide for System Administrators

Table of Contents

1. Overview

This document describes how you can customize the file smconfig.custom.xml to enhance Spectrum Mill to suit the needs of your laboratory. To fully understand this document, it is helpful to review the Spectrum Mill online help, especially the sections that involve the selection of modifications and the search modes available for MS/MS Search. You will also need a simple understanding of XML.

While the current Spectrum Mill release provides many capabilities to define custom modifications, there are still a few classes of modifications that require more than edits to smconfig files. Examples are modifications that require different enzymes for a search cycle, and those that remove an amino acid as part of the modification. The former is not supported, while the latter is exemplified by Acetyl, which removes a protein N- terminal methionine. The Spectrum Mill MS/MS Search program specifically checks for this modification in order to apply that rule. New types of isobaric reporter ion modifications beyond what is currently supported will also require a software update. As new chemistries are developed for labeling and modifying amino acids, expect that hard-coded changes to the Spectrum Mill programs may be required to support the modifications. In many cases, the special search cycle feature combined with a variable modifications search, or a search of saved results, can overcome a particular limitation.

1.1 Spectrum Mill Configuration Files

A default set of definitions for elements, amino acids, modifications, and homology search modes is contained in the file **smconfig.std.xml**. **Do not edit or modify smconfig.std.xml.** Instead, create and edit a file named **smconfig.custom.xml**, and place it in the **\SpectrumMill\msparams_mill** folder along with the **smconfig.std.xml** file. **Keep your custom modifications in a separate file. Future re-installs or upgrades of Spectrum Mill from may overwrite smconfig.std.xml but will not overwrite smconfig.custom.xml**.

When you click the **Choose…** button to select modifications on the Spectrum Mill forms, two things happen automatically:

1. If the file **smconfig.custom.xml** exists, it is merged with **smconfig.std.xml** to produce the **smconfig.xml** file that is used by Spectrum Mill programs.

2. The **smconfig.xml** file is used to generate a JavaScript file (**mods.js**) that is used to generate the list of modifications in the **Choose Modifications** dialog that you access from the Spectrum Mill forms.

When Spectrum Mill is installed, three additional smconfig xml files are installed:

File	Description
smconfig.misc.xml	Additional modifications that are provided "as-is", meaning they have not undergone testing by Spectrum Mill developers.
smconfig.custom.example.xml	An example of a smconfig.custom.xml file. *Do not use this file as your smconfig.custom.xml file.*
smconfig.custom.empty.xml	An empty smconfig.custom.xml file that contains the main (XML) elements that are necessary. You should save this file as **smconfig.custom.xml** and use it as the starting place for your own custom file.

You can use the **smconfig.custom.empty.xml** file as the starter for your **smconfig.custom.xml** file. The **smconfig.custom.example.xml** serves as a simple example of what can be done to customize your **smconfig.custom.xml** file. The **smconfig.misc.xml** files contains various element, modification, and homology search definitions that you may copy and paste into your **smconfig.custom.xml**. Please note, however, that the definitions in **smconfig.misc.xml** have not been thoroughly tested by Agilent. You should carefully review the definitions, and compare the definitions with the spectra you acquire from samples that contain those modifications, to be sure that the definitions are correct.

Note: Do not resave the smconfig.custom.example.xml file as smconfig.custom.xml. Instead, start with the smconfig.custom.empty.xml file that is provided, and save it as smconfig.custom.xml. When editing smconfig.custom.xml, you must use a text editor that does not add formatting or other embellishments to the text.

When you make changes to your **smconfig.custom.xml** file, it is good practice to save a backup copy. After you make changes, you should verify that you do not have any XML-related mistakes in your file. The simplest way to verify that the XML content is proper is to load the file into Internet Explorer. (Click **OK** to any warning about viewing active content.) Page down or scroll down to the bottom of the file and make sure that there are no errors. If there are errors in the XML, Internet Explorer indicates the line number and character position. You should also carefully review the definitions for the modifications, to make sure that they contain the proper elements and attribute values, as described in this document.

Within the Spectrum Mill Tool Belt, the option to **List modification details** allows any Spectrum Mill client to list the available modifications (including hidden ones) that are defined in **smconfig.xml**, **smconfig.std.xml**, or **smconfig.misc.xml**. The **Details…** button in the **Choose Modifications** dialog displays a list of all available modifications defined in the generated **smconfig.xml**. You can use these features to double-check that your changes are correct.

1.2 Quick Guide to Customizing Spectrum Mill

Note: A potential source of confusion is over the word "element" which is necessary for both XML and for chemistry. For example, chemical "elements" are defined using the XML "element" <element>. Where there may be confusion, we will refer to either (XML) elements or (chemical) elements.

1.2.1 Basics

The simplest way to customize your **smconfig.custom.xml** is to find a related <element> or <mod> definition in either **smconfig.std.xml** or **smconfig.misc.xml**. Open the files in a plain text editor such as Notepad, or in an XML editor. (Do not use a full-featured word processing program because it may introduce unwanted formatting.) Copy the definition from **smconfig.std.xml** or **smconfig.misc.xml** and paste it into the appropriate location within **smconfig.custom.xml**. Note that there are locations for the following:

- **Element definitions, under <elements>**
- **Modification definitions, under <modifications>**
- **Homology definitions, under <homology>**

Then edit the attributes and other contained (XML) elements to fit your definition. The rest of this section provides important tips. For more details, please click the links in the bulleted list above.

Note: As you read through this document, it is very helpful to locate examples in smconfig.std.xml, smconfig.misc.xml, and/or smconfig.custom.example.xml. These files are on your Spectrum Mill server, under

SpectrumMill\msparams_mill. Double-click a file to view it.

1.2.2 Tips

For (chemical) element (<element>) definitions, make sure the "symbol" attribute is unique, and that all of the isotopes (<isotope>) are properly defined. If you copy and paste an <element> definition, be sure to copy both the main tag (<element>) and everything through the closing tag (</element>).

For modification (<mod>) definitions, make sure that the "id" attribute is unique, and the "name" attribute has the proper text you wish to display. If you copy and paste a <mod> definition, be sure to copy both the main tag (<mod>) and everything through the closing tag (</mod>). The "formula" corresponds to the mass of the modification, that is, the total formula minus the unmodified amino acid formula. If the modification affects other amino acid attributes, such as pK values, you should specify those values. Attributes that are not specified are assumed to be the same as the unmodified amino acid. When defining a DEQ ("Differential Expression Quantitation") modification, you will need to define all of the modified forms, and the mix modification cycles. Finally, it is important to define any marker ions (<markerIon>) that are found in the spectra.

For new homology search modes (<mode>) definitions, again it is simplest to copy and paste an existing definition into your file, and then make changes. If you copy and paste a <mode> definition, be sure to copy both the main tag (<mode>) and everything through the closing tag (</mode>). Make sure the "id" attribute value is unique, and then define the mappings (<map>).

The XML element hierarchy is summarized under SMConfig Format, with links to the details for each element's descriptions within this document.

1.3 Using XML

XML has become a standard for formatting content so that it can be easily read by humans and interpreted (parsed) by software. It is similar to HTML, but with two key differences: every tag (element) must have a closing tag, and it is case-sensitive. XML is finding increasing use as a mechanism for storing configuration

data, including mass spectrometry data. Spectrum Mill uses XML for configuration information related to (chemical) elements, amino acids, modifications to amino acids, and MS/MS search modes.

You do not have to be an XML expert to create custom modifications or new homology search modes for Spectrum Mill. In fact, chemical and mass spectrometry knowledge is far more important when defining modifications. This section gives enough of an overview of XML to get you started, so that you can understand how XML is used in the Spectrum Mill configuration file.

XML documents are made up of two main concepts: elements and attributes. Elements are enclosed in angle brackets (e.g., <mod>), and may contain zero or more attributes. Elements must have a corresponding closing tag (e.g. </mod>). When elements to not contain other "inner" elements or text, the closing tag shortcut "/>" may be used (e.g, <site aa="K" />). XML attributes are special parameters that help define the meaning of the XML element. Attributes are expressed as an identifier="value". Here is an example:

<anXMLElement anAttribute="value1" anotherAttribute="2.04" >
Any text or other elements may be contained within the element, up until the closing element tag:
</anXMLElement>

The smconfig file does not make use of any inner text between element tags. For elements that contain only attributes, the element tag may be closed by adding a "/>" to the end:

<anXMLElementWithNothingElse />

A comment can be entered using special start (<!--) and end (-->) tokens:

<!-- this is a comment -->

When entering comments, do not use double dashes (--) anywhere in the comment, except for the start and end tokens.

XML documents are hierarchical—elements can contain other elements. Each document must have a root document element. For Spectrum Mill's smconfig file, this is the <configurations> element. The complete hierarchy for smconfig XML is given under SMConfig Format.

There is much debate over the use of elements alone or elements in combination with attributes. The Spectrum Mill configuration files make use of attributes, since it leads to smaller file sizes and is usually easier to understand. Another consideration is the trade-off between self-explanatory (often long) element and attribute names, versus conciseness. Spectrum Mill's XML configuration syntax leans towards conciseness, which improves performance at the expense of some readability. That is, we have tried to maintain reasonable information content-to-size ratio. The **smconfig.xml** file is loaded by extraction, search, and some result summary programs, and is transferred to the web browser when displaying spectra in the Spectrum Viewer.
Since the **smconfig.xml** file has been kept as small as possible, performance is faster during these critical operations.

1.4 SMConfig Format

The XML format for **smconfig.xml** is based on the following XML element hierarchy. Each of the above elements and their attributes are documented under SMConfig XML Specification. To view a file that uses this hierarchy, double-click **SpectrumMill\msparams_mill\smconfig.std.xml**.

XML Element	Description
<configurations>	Document root element
<config>	Defines a configuration
<elements>	Begins section for defining elements
<element>	Defines an element
<isotope />	Defines an element isotope
</element>	Ends an <element> definition
</elements>	Ends the <elements> section
<aminoAcids>	Begins section for defining amino acids
<aa>	Defines an amino acid
<markerIon />	Defines a markerIon for an amino acid
</aa>	Ends an <aa> definition
</aminoAcids>	Ends the <aminoAcids> section
<modifications>	Begins section for defining modifications
<mod>	Defines a modification
<site />	Defines a site (amino acid and/or termini) for a modification
<cycles>	Defines a set of search cycles (optional)
<cycle>	Defines a search cycle
<cyclemod />	Defines a modification to apply for a <cycle>
</cycle>	Ends a <cycle> definition
</cycles>	Ends a <cycles> definition
<markerIon />	Defines a makerIon for a modification
</mod>	Ends a modification definition
</modifications>	Ends the <modifications> section
<homology>	Begins the section for defining homology mappings
<mode>	Begins a section for defining a homology search mode
<map />	Defines a homology substitution mapping
</mode>	Ends a homology search mode
</homology>	Ends the <homology> section
</config>	Ends a <config> section
</configurations>	Ends the document root <configurations> section

2. SMConfig XML Specification

This section describes the details for each XML element and any attributes that are used to define a configuration in smconfig.

<configurations>

The <configurations> element is the root element for all XML elements in the file.

A <configurations> element may contain zero or more <config> elements. In this release, only one <config> element is currently supported, and the <config> element must have id="default".

Attributes: None.

Elements:

Element	Meaning	Required
config	Defines a configuration	1 or more (only 1 allowed in this release)

<config>

A <config> defines a set of elements, amino acids, modifications, and homology search mappings. A <config> contains zero or one <elements> (XML) elements, zero or one <aminoAcids> elements, zero or one <modifications> elements, and zero or one <homology> elements. Note that these (XML) elements are optional only in the **smconfig.custom.xml** file. The **smconfig.std.xml** and the auto-generated **smconfig.xml** must include all of these (XML) elements.

In this release, only one <config> element is allowed, and it must have id="default".

Attributes:

Attribute	Meaning	Required
id	Identifies the configuration. The id value must be unique amongst other <config> elements. *Note:* Only one <config> is supported in this release.	Yes (only 1 and must be "default")

Elements:

Element	Meaning	Required
elements	Contains a one or more <element> definitions	0 or 1
aminoAcids	Contains one or more <aa> elements used to define amino acids	0 or 1
modifications	Contains one or more <mod> elements used to define an amino acid modification	0 or 1
homology	Contains one or more <mode> elements used to define a homology search mode	0 or 1

2.1 Defining Elements

<config>
<elements>

The <elements> (XML) element contains one or more <element> definitions for (chemical) elements.

Attributes: None.

Elements:

Element	Meaning	Required
element	Defines an element	0 or more

<config>
<elements>
<element>

The <element> defines a (chemical) element to be used when calculating masses of formulas.

Attributes:

Attribute	Meaning	Required
symbol	The element symbol. The symbol may be multiple characters, but must not end in a numeric. Typically this is the chemical symbol for the element. For transition metals, where multiple oxidation states are possible, use 'p' or some other character after the numeric. For example, Fe^{2+} could be represented as "Fe2p".	Yes

Elements:

Element	Meaning	Required
isotope	Defines a isotope for an element	1 or more

<config>
<elements>
<element>
<isotope>

One or more <isotope> (XML) elements define a (chemical) element. A (chemical) element is defined by its isotopes, which indicate the mass and the abundance. For monisotopic masses, Spectrum Mill uses the most abundant isotope, which must be listed first. For average mass calculations, the abundance of each isotope is considered.

Attributes:

Attribute	Meaning	Required
mass	The atomic mass of the isotope	Yes
abund	The isotope abundance expressed as a fraction of 1. For example, an isotope that has an abundance of 1% would have a value of abund="0.01".	Yes

Elements: None.

2.2 Defining Amino Acids

<config>
<aminoAcids>

The <aminoAcids> section defines amino acids (<aa> elements). The amino acids within **smconfig.std.xml** must not be modified, nor should they be replaced by custom definitions in **smconfig.custom.xml**. Instead, define new modifications that can be selected as fixed modifications.

Attributes: None.

Elements:

Element	Meaning	Required
aa	Defines an amino acid	1 or more

<config>
<aminoAcids>
<aa>

The <aa> element defines an amino acid. In general, there is no need to define new amino acids. Instead, you should define a modification that can be selected to apply to one or more amino acids. The main reason there is no need to define new amino acids is that the amino acid symbol must be one that would normally appear in a protein database sequence, which limits the symbols to those for the standard amino acids, plus X and Z. If you need to search for the latter, you can define a homology mapping that maps those to one or more standard amino acids or modifications.

Note that the standard amino acid definitions cannot be modified—the <aminoAcids> section in **smconfig.custom.xml** is ignored.

Attributes:

Attribute	Meaning	Required
symbol	The amino acid symbol (must be a single character).	Yes
id	The internal id used by the Spectrum Mill program to identify the amino acid. The id value must not contain spaces.	Yes
formula	The molecular formula for the amino acid. Amino acid masses are calculated using this formula.	Yes
AChain	The formula for the amino acid A chain. If there is no A chain (glycine, for example), the value should be “0”.	Yes
BChain	The formula for the amino acid B chain. If there is no B chain, the value should be “0”.	Yes
pKCterm	The pK value for the C-terminus of the amino acid	Yes
pKNterm	The pK value for the N-terminus of the amino acid	Yes
pKAside	The pK value of the A chain of the amino acid. If there is no A chain, set pKAside=“”	Yes
pKBside	The pK value of the B chain of the amino acid. If there is no B chain, set pKBside=“”	Yes

The following attributes are currently used only by the Spectrum Viewer and Sherenga *de novo* Sequencing. They may be used by other search programs in future versions of the Spectrum Mill workbench. These attributes are treated as "flags". Simply setting the value to "" is enough to set the condition to true, while absence indicates false.

Attribute	Meaning	Required
basic	The presence of this attribute indicates that the amino acid is basic.	Omit if not basic.
nh3loss	The presence of this attribute indicates that the amino acid is may lose NH_3 upon fragmentation.	Omit if no NH_3 loss is possible.
phos	The presence of this attribute indicates that the amino acid may lose a phosphate upon fragmentation.	Omit if phosphate losses are not possible.

Elements:

An amino acid definition can contain definitions for one or more marker ions. The typical marker ion is an immonium ion formed by the loss of CO. The definition of marker ions using the <markerIon> element is described later in this document.

Element	Meaning	Required
markerIon	Defines a marker ion for the amino acid. See the definition for <markerIon> below.	No. May have 0 or more.

<config>
  <aminoAcids>
    <aa>
      <markerIon>

The <markerIon> element defines a marker ion for the amino acid. Amino acid marker ions are generally low mass immonium ions formed by the loss of CO. Please see the <markerIon> section below for details about <markerIon> definitions.

2.3 Defining Modifications

<config>
  <modifications>

The <modifications> section allows you to define modifications that can be selected for consideration during a search. Modifications are defined using the <mod> element. In the **Choose Modifications** dialog that you access from the Spectrum Mill forms, fixed modifications for the same amino acid are listed in the order that they are defined in the <modifications> section. Variable modifications are listed in the order they appear in **smconfig.xml** (not grouped by amino acid site).

Attributes: None.
Elements:

Element	Meaning	Required
mod	Defines a modification.	0 or 1

<config>
  <modifications>
    <mod>

Fixed and variable modifications

A modification is defined using the <mod> element. A <mod> element contains one or more <site> elements that indicate an amino acid or terminus that is modified. The <mod> "type" attribute determines how the modification appears in the

Choose Modifications dialog that you access from the Spectrum Mill forms, and how it is applied during a search. Modifications fall into two categories: fixed and variable. Fixed modifications (type="fixed") redefine one or more amino acids or termini, and apply to all of the amino acids or termini that are indicated by the <site> elements. Variable modifications (type="variable") are considered as possible modifications during a search, where both modified and unmodified sites are tested when attempting to match a peptide. If a modification may be either fixed or variable, and you want to be able to select it under either list in the **Choose Modifications** dialog, set type="any".

Cyclic modifications

The Spectrum Mill MS/MS Search supports "cyclic modifications", which trigger "search cycles". (Note that "cyclic modifications" do not imply that the chemistry of the modification is "cyclic".). When type="cyclic", then the modification triggers multiple searches, where a different modified form (e.g., ICAT-D_0 or ICAT-D_8) is searched in each cycle.

While most cyclic search modifications are used in differential expression quantitation (DEQ) searches, it is not required that they be used that way. It is useful to define a cyclic search modification any time a modification is either always present on all indicated sites, or is not present on any site. That way, one search cycle can look for the presence of the modification on all indicated sites, while a second search cycle can look for the absence of the modification on all indicated sites. This is unlike a variable modification, where both modified and unmodified sites can be present within a given search cycle on a given peptide.

To define a cyclic modification, first define the fixed modifications that you want to search in each of the search cycles. Then define the cyclic modification (type="cyclic"). Within **smconfig.std.xml**, the definitions for cICAT-C12, cICAT-C13, and cICAT-mix provide good examples. By convention, the id and name attributes for cyclic modifications end in "-mix", but this is not required. Note that the modifications you specify within the search cycles must be fixed modifications (type="fixed" or type="any"). For example, cICAT-C12 and cICAT- C13 are fixed modifications, while cICAT-mix is the cyclic modification.

A "DEQId" attribute is used to group modifications that are used for differential expression quantitation. For example, the cICAT DEQ modification is defined using DEQId="cICAT", and by defining a cICAT-C12 modification, a cICAT-C13 modification, and a cICAT-mix modification that specifies a search cycle for cICAT-C12 followed by a search cycle for cICAT-C13.

The DEQType attribute indicates the type of DEQ modification. The values may be "normal" for ICAT-like modifications, "SILAC" for SILAC-type modifications, and "isotope" for metabolic isotope modifications where all amino acids are affected by the isotope substitution.

Isobaric repoter ion modifications (iTRAQ, TMT) may also be searched in a "mix" cycle. For example, the standard iTRAQ modification is searched in one cycle, and the type is "fixed". The iTRAQ Partial-mix modification checks for incomplete labeling and searches in four cycles: no label, lysine-only label, N-terminal- only label, and complete label (both lysines and N-terminus).

Hiding modifications

To prevent a modification from being shown in the **Choose Modifications** dialog, set the "hide" attribute to "1". In some cases, it is either not necessary or permissible for a user to be able to select a particular modification that is part of a cyclic search. In other cases (for example, for backwards compatibility with Spectrum Mill version A.03.01 and earlier), the modification needs to be defined for summary reports, but is not to be used for new searches. The hide= "1" attribute is appropriate for these situations.

Attributes:

Attribute	Meaning	Required
id	Internal identifier for the modification. The value must be unique among all <mod> elements. Spaces are not permitted.	Yes
name	The name that is displayed in the user interface and reports. Spaces are permitted.	Yes
formula	The delta formula for the modification. The calculated formula mass is added to the mass of the site.	Yes, except it is omitted for modifications that define search cycles.
type	The type of modification, which may be "fixed", "variable", or "any". (See Fixed and variable modifications.)	Yes
DEQId	The "base id" that is in common for the set of <mod> elements that define a DEQ modification.	Only for DEQ modifications
DEQ	Identifies the modification as being a DEQ modification and indicates the search cycle where it is applied. The value is a number indicating the search cycle. Example: DEQ="1", DEQ="2", etc. The value is "M" for the modification that defines the "-mix" search cycles.	Only for DEQ modifications
hide	Prevents a modification from being displayed in the **Choose Modifications** dialog. You may either use hide="" or hide="1". To show a modification that is hidden in **smconfig.std.xml**, set hide="0" for the attribute in your smconfig.custom.xml file.	Optional
PMFScoring	This attribute determines how the (variable) modification is to contribute to scoring in PMF searches. Possible values are: "0": do not score if match found "1": always score if match found "2": only score if unmodified match is also found	Optional, default is "0". Applies only to type="variable" or "any".
abbrev	Abbreviation for the (variable) modifications when reporting PMF search results.	Optional, but should be set for variable modifications used in PMF searches.
useMetabolicIsotope	Use this attribute when the formula contains a potential isotope used in a metabolic isotope search (such as ^{15}N-mix). If present, the current isotopic element form in the search cycle will be used when calculating the delta mass from the formula.	Optional, depends upon modification.
metabolic	Use this attribute to define a modification that occurs when growing cells in an isotopic media. Any variable modification to a fixed modification site that has metabolic set will modify the mass of the fixed modification rather than replacing it. For example, SILAC modifications include this attribute.	Optional, depends upon modification.
iTRAQ	This attribute was used for isobaric reporter ion modifications, such as iTRAQ and TMT. This attributes was applied to indicate that **markerIon** masses were to be calculated.as absolute formulas. As of B.06.00, the **markerIon** "absoluteFormula" attribute is to be used instead.	Deprecated. Use **markerIon** 'absoluteFormula' attribute instead.
isobaric	This attribute was used for isobaric reporter ion modifications, such as iTRAQ and TMT and replaced the use of the "iTRAQ" attribute. This attributes was applied to indicate that **markerIon** masses were to be calculated.as absolute formulas. As of B.06.00, the **markerIon** "absoluteFormula" attribute is to be used instead.	Deprecated. Use **markerIon** 'absoluteFormula' attribute instead.

Elements:

A modification (<mod>) definition can contain definitions for one or more marker ions (<markerIon>) and one or more modification sites (<site>). A modification (<mod>) definition may also define a set of search cycles (<cycles>).

Typically, the marker ions that are common for the indicated unmodified amino acid should be defined for the modification. In addition, some modifications, especially those with large side chains such as ICAT, may require that you define additional marker ions. For details, please see the <markerIon> definition section later in this document.

MS/MS Search supports the automatic search of multiple fixed modifications. These cyclic searches are most often used in DEQ searches, but have other uses, as described above under Cyclic modifications. You define a modification to indicate a cyclic search by defining the search cycles (<cycles>).

Element	Meaning	Required
site	Specifies an amino acid or a terminus that is the target of this modification.	Yes, 1 or more.
markerIon	Defines a marker ion for the amino acid. See the definition for <markerIon> below.	Optional, but should be defined if expected to be seen in spectra.
cycles	Defines a set of search cycles (<cycle>). Only MS/MS Search (not PMF Search) supports search cycles.	No, but only one allowed.

<config>
<modifications>
<mod>
<site>

A <site> element indicates which amino acid or terminus is modified by the modification. You specify multiple sites by defining multiple <site> elements within the <mod>, or by specifying multiple amino acids in the "for" attribute. Note that if a modification modifies only a terminus, it must be defined as fixed (type="fixed"). The exception to this rule is if the modification only modifies a terminus when a specific amino acid is at the terminus (PyroGlu and Q, for example). These amino-acid-specific-termini modifications are supported by searches.

When you define a cyclic search modification (type="cyclic"), you must specify all sites that are affected by the individual cycles.

Attributes:

Attribute	Meaning	Required
aa	Specifies one or more amino acids that may be modified. A "*" indicates any amino acid may be modified. The "*" value is typically used for termini modifications, or for general metabolic isotope modifications (such as ^{15}N).	Yes.
term	Specifies the terminus site. Possible values are: "n" – N-terminus "c" –C-terminus "^" – Protein N-terminus "$" – Protein C-terminus When the modification occurs at a terminus, the aa attribute typically contains a "*", but not always. Some modifications (such as PyroGlu) only modify a terminus if a specific amino acid is the terminus (Q for PyroGlu).	Only if modification applies to a terminus.
ref	If present, the value refers to another modification that is to be used to define the modification for the site. This attribute is used to define a modification that is a combination of others. For example, **smconfig.std.xml** defines various "PTM–" modifications for backwards compatibility of results that were searched with earlier versions of the Spectrum Mill workbench. (Search for **PTM** to find these examples.)	No.
AChain BChain pkAside pkBside basic nh3loss phos	If specified, these values override those for the unmodified amino acid. See the <aa> section above for the meaning of these attributes.	Optional.

Elements:

Element	Meaning	Required
isotope	Defines an isotope for an element	1 or more

<config>
<modifications>
<mod>
<markerIon>

While marker ions are typically site-specific, they are not defined within <site> elements. The <markerIon> "for" attribute indicates the sites that support a particular marker ion for a modification. The <markerIon> element is defined later in this document. Please see that section for details.

<config>
<modifications>
<mod>
<cycles>

Defines a set of search cycles (<cycle> elements). The type attribute for the modification must be set to "cyclic".

Attributes: None.

Elements:

Element	Meaning	Required
cycle	Defines a specific search cycle.	2 or more

<config>
<modifications>
<mod>
<cycles>
<cycle>

A <cycle> element defines a search cycle. The modifications to apply during a cycle are defined using the <cyclemod> element. If there are no modifications for a cycle, the <cyclemod> is omitted. A <cycle> contains an attribute, "n" which indicates the cycle number, and serves as an identifier.

Attributes:

Attribute	Meaning	Required
n	Indicates the search cycle number.	Yes

Elements:

Element	Meaning	Required
cyclemod	Defines a modification to be used for the search cycle. Only one <cyclemod> per cycle is supported in this release.	0 or 1

<config>
<modifications>
<mod>
<cycles>
<cycle>
<cyclemod>

A <cyclemod> element indicates the modification to be applied for the cycle. Only fixed (type="fixed" or type="any") modifications may be indicated. If a cycle step does not apply any modifications, the <cyclemod> element is omitted. In this release, only one <cyclemod> element per <cycle> is supported.

Attributes:

Attribute	Meaning	Required
mod	Assigns the modification to be applied for the cycle. The value is a <mod> id value for a fixed modification (type="fixed" or type="any").	Yes

Elements: None.

<markerIon>

A <markerIon> element defines a marker ion. The <markerIon> element is used to define marker ions for both amino acids (<aa>) and modifications (<mod>).

The most common type of marker ion is an immonium ion formed by the loss of CO during fragmentation. These ions indicate a specific amino acid composition, and thus are sometimes called composition ions.
Immonium ions are generally found in the low mass range, and so appear only in spectra from instruments that can measure in the low mass (< 250 amu) range. Marker ions can also result from a neutral loss from the parent ion. The most common examples are the marker ions associated with phosphate losses from phosphorylated serine, threonine, or tyrosine. Finally, bulky modifications that create a large side chain on an amino acid residue may form both a neutral loss from the parent and one or more ionized markers of constant mass. The ICAT modification is a good example.

Because marker ions can give intense peaks that can interfere with the normal interpretation of b/y pairs, both MS/MS Search and *de novo* Sequencing shrink the intensities of these marker ions while searching. If a marker ion is identified, it is scored according to the "bonus" and "penalty" attribute values. In a spectrum, if multiple marker ions with the same set of "for" amino acids match peaks, only one of those markers will contribute to the score. The order of the marker ions does not have to match, just the set. For example, markers for="KQ" and for="QK" will score only once. The phosphorylated-S markers are another example. Even if all three "st" marker ions are found, only one is scored. If found, the "sty" marker ion is scored even if an "st" marker ion is also found.

The <markerIon> element contains two important attributes that determine how the mass of the marker ion is calculated: the "ionized" and "pmni" attributes. "pmni" stands for "parent minus neutral ionized". In all cases, the marker ion formula must not include the cation mass (that is, do not add an extra "H" to the formula)—the cation mass is determined when you run the search, based on the mass type that is selected for the search.

Marker ion type	ionized	pmni	Mass calculation	Example
Ion formed by loss from amino acid	1	0	The formula mass is subtracted from the amino acid mass.	Immonium ion
Ion formed by neutral loss from parent ion	0	1	The formula mass is subtracted from the parent ion mass.	Phosphorylation
Two ions are formed: one by loss from the amino acid, the other by loss from the parent ion	1	1	The formula mass represents the mass of the ionized marker ion (minus the cation mass). This mass is subtracted from the parent ion to form the second marker ion.	ICAT

If you fail to properly define marker ions that appear in spectra, you may identify significantly fewer peptides in a search. For most modifications, you can define marker ions by simply copying the <markerIon> definitions from the unmodified amino acid, and adjusting the formulas for any isotopes that are part of the modification. If the modification adds significantly to a side chain, and is likely to fragment or "fall off" while carrying a charge,

then define a <markerIon> with ionized="1" and pmni="1". Keep in mind that the formula mass will be "one less" than the nominal mass for the ion—Spectrum Mill will automatically add the proper mass (proton or hydrogen) based on the mass type selected for the search.

The "nMass" (nominal mass) attribute is not used, but is useful for documenting the marker ions. You should add 1 to the formula mass to indicate the nominal mass.

Attributes:

Attribute	Meaning	Required
for	Indicates the amino acids that can form a marker ion of this mass. If multiple amino acids can form the same marker ions of the same mass, each should be listed, and each amino acid should have its own <markerIon> definition for that mass. The first amino acid listed is the one used to calculate the overall mass. (The overall mass is the amino acid mass minus the marker ion mass.) For marker ions associated with variable modifications, the list of amino acids should be lower-case, e.g. "sty". If multiple marker ions have the same set of "for" attributes, only one of the marker ions contributes to the score. The "for" attribute value is reported as the marker ion label in the MS/MS Search file details report.	Yes
label	The text that is displayed in the Spectrum Viewer. The label may be in HTML format (see the marker ions for phosphorylation for examples).	Yes
formula	Specifies the formula used to calculate the marker ion mass. An empty value (formula="") indicates that ion is the amino acid mass plus a proton. How the formula is specified depends upon the ionized and pmni attribute values. See the above table for details. The "absoluteFormula" attribute is used to indicate that the mass is that specified by the formula.	Yes
absoluteFormula	Inidcates (value of "1") that the marker ion formula specifies the mass of the marker ion, regardless of the ionized and pmni settings. The "Lys-only" iTRAQ marker ions are an example where it must be set.	Optional
ionized	Indicates (value of "1") if the marker ion is formed by a loss from an amino acid. If the ion is formed only by a neutral loss from the parent (pmni="1" as well), then set ionized="0". See the above table for details.	Yes
pmni	"parent minus neutral ionized" indicates that the ion is formed by a neutral loss from the parent ion. See the above table for details.	Yes
bonus	The bonus score given to the peptide if the marker ion is found. (If multiple marker ions have the same set of "for" amino acids, only the largest score counts.)	Yes
penalty	The amount to be subtracted from the score if the marker ion is not found in the spectrum	Yes
hiCID	If present, the marker ion is only enabled if the instrument supports high energy CID (hiEnergyCID value in instrument.txt). To flag a marker ion as only high energy CID, set hiCID="1".	Must be present to set flag.
nMass	The "nominal mass" of the marker ion. This attribute is for documentation hints only. The value should be 1 more than the formula mass.	No, but recommended.

Elements: None.

2.4 Defining Homology Search Modes

<config>
<homology>

The <homology> element contains the section for defining homology search modes. It is possible to define custom homology search modes. Homology search modes specify a substitution mapping for one or more amino acids. When searching in a homology mode, each variation (mapping) of an amino acid defined in the homology
<mode> <map> is considered, subject to mass range restrictions. Only one substitution is allowed in the peptide. When variable modifications are selected as part of the search, they are added to the possible substitution mappings.

Attributes: None.

Elements:

Element	Meaning	Required
<mode>	Defines a homology search mode.	0 or more.

<config>
<homology>
<mode>

The <mode> element defines a particular homology search mode that can be selected in MS/MS Search. It contains one or more <map> elements, which define which substitutions can be attempted.

Attributes:

Attribute	Meaning	Required
id	Internal id for the homology search mode. The id value must be unique among all other <mode> elements, and must not contain spaces.	Yes
name	The text that is displayed in the MS/MS Search mode menu.	Yes

Elements:

Element	Meaning	Required
map	Defines a substitution mapping for an amino acid that is found in the protein database.	1 or more

<config>
<homology>
<mode>
<map>

The <map> element defines a substitution mapping of an amino acid to other possible amino acids. The “aa” attribute is the amino acid symbol that is to be replaced, and the “to” attribute lists the amino acids that may be substituted for the “aa” amino acid. The amino acid symbols are those that can be found in a protein database, and do not necessarily have to be a standard amino acid. For example, the symbol “Z” may be mapped to one or more amino acids using a homology map mode. If you select variable modifications for the homology mode search, the variable modifications are appended to the “to” list as possible substitutions. Note that homology searches allow only one substitution per peptide.

Attributes:

Attribute	Meaning	Required
aa	The amino acid, indicated by a single character symbol, that is to be mapped (replaced).	Yes
to	A list of amino acids, indicated by a list of single character symbols, that are to be considered as possible substitutions for the "aa" amino acid.	Yes

Elements: None.

2.5 Special Attributes for smconfig.custom.xml

You create custom modifications by creating and editing the file **smconfig.custom.xml** and saving it in **\SpectrumMill\msparams_mill**. You can do the following within your custom file:

- Create new (chemical) element definitions
- Create new modification definitions
- Override modification definitions that are defined in **smconfig.std.xml**
- Show or hide modification definitions that are defined in **smconfig.std.xml**
- Create new homology search modes
- Show or hide homology search modes that are defined in **smconfig.std.xml**

Note that you cannot use **smconfig.custom.xml** to modify or remove the standard amino acid definitions that are present in **smconfig.std.xml**.

The **smconfig.custom.xml** file is automatically merged with the **smconfig.std.xml** file to generate the **smconfig.xml** file used by the Spectrum Mill programs and the Spectrum Viewer.

The following are special "processing" attributes that affect how the **smconfig.custom.xml** definitions are merged with those in **smconfig.std.xml**:

Attribute	Meaning
REMOVE	The definition will be removed and not merged into **smconfig.xml**.
INSERT-AT	Specifies where to insert the definition. Possible values are: "begin" – insert at beginning of section "end" – insert at end of section "*id*" – insert just before the definition with that *id* The INSERT-AT directive is most useful for modifications (<mod>) where the order of appearance in **smconfig.xml** determines the order that the modification is listed in the **Choose Modifications** dialog. The default is "end" if not specified.
hide	Applies only to modifications (<mod>) and homology modes (<mode>). A value of "1" prevents the modification or search mode from being shown in the user interface. A value of "0" is used to show a modification that is defined as hidden (hide="1") in **smconfig.std.xml**.

Hidden modifications can be used by searches and the Spectrum Viewer on previously searched data. Users are not able to select them for new extractions and/or searches. Note that the INSERT-AT and REMOVE attributes are all upper case, while the "hide" attribute is all lower case.

When you simply want to hide or show a modification, you only have to define the <mod> with the id attribute and the hide attribute. For example, the following enables display of the "PTM-KMQSTY" modification, which is defined to be hidden in **smconfig.std.xml**:

<mod id="PTM-KMQSTY" hide="0" />

(Note how the above uses the simplified tag close "/>" for the <mod> XML element.)

The REMOVE attribute also requires only the id plus the REMOVE attribute. The following removes all of the ICAT-mix related modifications:

<mod id="ICAT-D0" REMOVE="" />
<mod id="ICAT-D8" REMOVE="" />
<mod id="ICAT-mix" REMOVE="" />

The REMOVE attribute is useful to reduce the size and clutter of the generated **smconfig.xml** by removing modifications that your laboratory does not use. Please be aware when you remove modifications that your users may not be able to complete certain examples in the *Quick Start Guide*.

If you need to do more than hide, unhide, or remove a definition, you must specify the entire definition. When you specify an entire definition, the modification is redefined to be exactly what you define in **smconfig.custom.xml**. By using the INSERT-AT attribute, you can specify where the modification appears in the **Choose Modifications** list. The default is at the end of the list of modifications.