

jWM Army File format

Introduction

jWM uses XML to define army lists.

At the current stage in development the army list files are stored with the suffix “.xml”.

To be recognized they have to be placed in the same folder as the application.

There is also a DTD file which is supplied by me and should not be changed for compatibility reasons. It must be located in the same folder where the army lists and the application reside.

XML

XML files use tags to structure and identify the information.

A tag can have attributes. The value of an attribute must be enclosed in “ even if the value is a number.

A tag can consist of start and end tag enclosing additional information which can be text or other tags.

Tags included in other tags must start and end within an enclosing tag. They must not start inside a tag and end outside.

XML is case-sensitive so a Tag called A is different from a tag called a.

<code><WM:Tag /></code>	is just a tag the slash at the end means start and end tag combined into one.
<code><WM:Me gender = “m” age = “46” /></code>	is tag with two attributes.
<code><WM:Me gender = “m” age = “46”></code> I was born <code><WM:Adr></code> I live where I live <code><WM:/Adr></code> <code></WM:Me></code>	is tag with two attributes and has embedded text and an embedded attribute with embedded text.

DTD

The DTD file holds information that tells the application how the XML file should be structured. It also defines default values for attributes.

Generally you can use XML files without DTD files. In the case of jWM the DTD file is required, because it specifies default values if certain tags are omitted. It can also be used by XML editors to check the syntax of an army list. Please do not change the DTD file.

This DTD has to be referenced by the army lists. It is important to note that the URL for the DTD has no path part, just the name.

So the header part of an army file looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WM:Race SYSTEM "Race.dtd">
```

Looking at the DTD will tell you all about the syntax of the army list file but will tell you only so much about the semantics.

jWM Army File format

Tags in an army list

Every tag in an army list has an id attribute. The value of an id attribute must not have any spaces. The id must be unique within the file. No other tag must have an id with the same value. But with one exception the ids can be reused in other files.

I usually use only upper case characters for ids and use a prefix depending on the tag:

C_ for WM:Character and WM:CharacterUpgrade tags, U_ for WM:Unit and WM:UnitUpgrade tags, O_ for WM:Option and WM:ExcOption tags, R_ for WM:Rule tags and L_ for WM:Limit tags.

This is not a necessity.

```
<WM:Character id="C_GENR" name="General" type="General" atk="+2" cmd="9" pts="125" options="O_PORT" />
```

```
<WM:Character id="C_LED" name="Leader" type="Leader" atk="+1" cmd="8" pts="80" />
```

All tags can have body text. This text is treated as an info text. At the moment the body text of WM:Race, WM:Rule, WM:Limit and WM:CompLimit tags are displayed.

In the body text the character series \n can be used to force a new line. A \n at the end of a rule's body text forces the next rule to start at a new line.

A \b will toggle bold characters on and off. \i switches italic on and off.

\n, \b and \i are up to now only supported in rule body texts.

WM:Race

Each XML file has one Document Tag which is the „outmost“ tag enclosing all other tags.

In this case it is WM:Race. A file with this tag looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WM:Race SYSTEM "Race.dtd">
<WM:Race id="SELE350" name="Seleucids 350BC" armyName="Seleucid army 350BC">
  <WM:Character id="C_LED" name="Leader" type="Leader" atk="+1" cmd="8" pts="80" />
  :
</WM:Race>
```

If there is more than one WM:Race tag in a file only the first one will be used.

The WM:Race tag has the following attributes:

Name	Default	
id	no default	The unique identifier. No other race, even in other files, must have the same identifier.
name	no default	The name of the race i.e. „Celts“.
armyName	no default	The name of an army i.e. „Celtic army“.

The body text will be displayed when the nation info button is clicked.

jWM Army File format

WM:Character and WM:CharacterUpgrade

Each WM:Character tag describes one entry in the army list that has a character profile. Each WM:Character tag will be displayed in the jWM list of possible units.

A WM:CharacterUpgrade has exactly the same format as the WM:Character, but it will not appear in the list of possible units.

You use WM:CharacterUpgrade tags to define upgrade options for characters or mounts.

A WM:CharacterUpgrade is a stand alone tag, so even if it represents an upgrade for an WM:Character tag, it has to have all needed attributes and all options and rules have to be defined even if they are identical to those of the character being upgraded. Following the same logic the cost of an upgrade is not the upgrade cost, but the total cost of an upgraded unit.

Name	Default	
id	no default	The unique identifier.
name	no default	The name of the character i.e. „General“.
type	no default	One of the following: General, Subordinate, Leader, Hero, Wizard, Shaman, Mount or Upgrade. Upgrade is only used in special cases like the Qurubiti. A WM:CharacterUpgrade normally has the same type as the WM:Character tag it upgrades or Mount in case of a mount. Will be displayed in the Characters profile.
atk	no default	The number of attacks i.e + .
rng	“-”	The range of any ranged weapons.
cmd	no default	Command value
siz	“1”	Size value
pts	no default	Points cost
options	no default	This is a list of space delimited ids. Like: “PORT CHAR”. If the id specifies a WM:CharacterUpgrade tag. A checkbox will be added which allows the addition of this CharacterUpgrade. Typical use is a mount, if only one type of mount is available. If the id specifies a WM:Option tag. A checkbox will be added which allows the addition of this Option. Typical use is a single magic item choice or anything else that has no profile. If the id specifies a WM:ExcOption tag. A combobox will be added which allows to choose one of the options defined in the WM:ExcOption tag. See it’s description for more details.
upgrades	no default	This is a list of space delimited ids. Like: “ALEX”. The ids specify WM:CharacterUpgrade tags. A combobox will be added which allows to choose one of the upgrades. WM:CharacterUpgrade tags do not have an upgrade attribute.

jWM Army File format

Name	Default	
rules	no default	This is a list of space delimited ids. Like: "PHALANX SHOCK". The ids specify WM:Rule tags that define the special rules valid for the character.
pera	"0"	Number of characters of this type added when a new army is created. The number of characters to add on behalf of a WM:Character Tag does not reflect the number of characters to be added on behalf on it's WM:CharacterUpgrade Tags.
perk	"0"	Number of characters of this type per 1000 points added when a new army is created. The number of characters to add on behalf of a WM:Character Tag does not reflect the number of characters to be added on behalf on it's WM:CharacterUpgrade Tags.
break	"0"	If set to „1“, a character of this type counts towards the entries for the armies breakpoints. If set to „0“ it doesn't. For units break is usually 1.
hide	"no"	Yes or no. If this attribute is yes the profile will not be shown. This is useful when an option is needed that does not display anything, like a No Mount option. WM:Character tags do not have an hide attribute.

The body text can hold a description of the character. This description will neither be displayed nor printed yet.

WM:Unit and WM:UnitUpgrade

Each WM:Unit tag describes one entry in the army list that has a unit profile. Each WM:Unit tag will be displayed in the jWM list of possible units.

A WM: WM:Unit Upgrade has exactly the same format as the WM:Unit, but it will not appear in the list of possible units.

You use WM: WM:Unit Upgrade tags to define upgrade options for units or sub units like skirmishers in the old Empire list.

A WM:UnitUpgrade is a stand alone tag, so even if it represents an upgrade for an WM:Unit tag, it has to have all needed attributes and all options and rules have to be defined even if they are identical to those of the unit being upgraded. Following the same logic the cost of an upgrade is not the upgrade cost, but the total cost of an upgraded unit.

Name	Default	
id	no default	The unique identifier.
name	no default	The name of the character i.e. „Phalanx“.

jWM Army File format

Name	Default	
type	no default	One of the following: Infantry, Cavalry, Chariot, Artillery, Elephant, Monster or Upgrade. Upgrade is only used in special cases like the Qurubiti. A WM:UbitUpgrade normally has the same type as the WM:Unit tag it upgrades. Will be displayed in the Units profile.
atk	no default	The number of attacks i.e + .
rng	"_"	The range of any ranged weapons.
hit	"3"	Number of hits value
arm	"_"	Armor value,
siz	"3"	Size value
pts	no default	Points cost
options	no default	This is a list of space delimited ids. Like: "SKIRM MAGIC". If the id specifies a WM:UnitUpgrade tag. A checkbox will be added which allows the addition of this UnitUpgrade. Typical use are Skirmishers in the old Empire list. If the id specifies a WM:Option tag. A checkbox will be added which allows the addition of this Option. Typical use is a single magic item choice or anything else that has no profile. If the id specifies a WM:ExcOption tag. A combobox will be added which allows to choose one of the options defined in the WM:ExcOption tag. See it's description for more details.
upgrades	no default	This is a list of space delimited ids. Like: "HYPA IMIT". The ids specify WM:UnitUpgrade tags. A combobox will be added which allows to choose one of the upgrades. WM:UnitUpgrade tags do not have an upgrade attribute.
rules	no default	This is a list of space delimited ids. Like: "PHALANX SHOCK". The ids specify WM:Rule tags that define the special rules valid for the character.
pera	"0"	Number of units of this type added when a new army is created. The number of units to add on behalf of a Unit Tag does not reflect the number of units to be added on behalf on it's WM:UnitUpgrade Tags.
perk	"0"	Number of units of this type per 1000 points added when a new army is created. The number of units to add on behalf of a Unit Tag does not reflect the number of units to be added on behalf on it's WM:UnitUpgrade Tags.

jWM Army File format

Name	Default	
break	"1"	If set to „1“, a unit of this type counts towards the units for the armies breakpoints. If set to „0“ it doesn't. If set to „s“ the unit only counts towards the breakpoint, if arm is not equal „-“, or if at least half of all units have a break of „s“. For units break is usually 1.
hide	"no"	Yes or no. If this attribute is yes the profile will not be shown. This is useful when an option is needed that does not display anything, like a No Skirmisher option. WM:Unit tags do not have an hide attribute.

The body text can hold an unit description. This description will neither be displayed nor printed yet.

Note that there are no min and max fields. All limits and composition tests are controlled and defined by WM:Limit and WM:CompLimit tags.

jWM Army File format

WM:Option

Each WM:Option tag describes one option that has no profile. Like a magic item, or an weapon option.

Each WM:Option tag describes a single option and if its id is part of an options attribute of a WM:Character/WM:CharacterUpgrade/WM:Unit/WM:UnitUpgrade it will be displayed as an checkbox.

If more than one option are available to an entry you just add their ids to the options attribute. If some options are exclusive, you add an WM:ExcOption to the options attribute and list all exclusive options in the options attribute of the WM:ExcOption tag.

Name	Default	
id	no default	The unique identifier.
name	no default	The name of the option i.e. „Dispel Scroll“.
pts	no default	Points cost
rules	no default	This is a list of space delimited ids. Like: “SCROLL MAGIC”. The ids specify WM:Rule tags that define the special rules valid for the option.

The body text can hold an option description. This description will neither be displayed nor printed yet.

WM:ExcOption

Each WM:ExcOption tag groups options that are exclusive to each other. Like nop mout, a chariot mount and an elephant mount.

Name	Default	
id	no default	The unique identifier.
name	no default	The name of the option i.e. „Mount“.
pts	no default	Points cost
options	no default	This is a list of space delimited ids. Like: “NOMOUNT CHARIOT ELEPHANT”. The ids specify WM:Option, WM:UnitUpgrade or WM:CharacterUpgrade tags that define the options which are grouped under this WM:ExcOption tag.
rules	no default	This is a list of space delimited ids. Like: “SCROLL MAGIC”. The ids specify WM:Rule tags that define the special rules valid for the option. Those rules are only displayed when the ExcOption is talked about, this is only the case when the „generic“ character or unit is displayed. The rules listed here are not displayed for any real unit. For real units the special rule of the selected option are shown.

The body text can hold an option description. This description will neither be displayed nor printed yet.

jWM Army File format

WM:Rule

Each WM:Rule tag describes one rule. This can be a units, or an items rule or whatever,

Name	Default	
id	no default	The unique identifier.
name	no default	The name of the rule i.e. „Shock“.

The body text can hold a description of the rule. This description will be displayed when a unit affected by this rule is displayed.

If a rule has no body text it's name is still displayed with any unit affected.

WM:Limit

Each WM:Limit tag describes one Test which has to be passed for the army list to be correct.

Name	Default	
id	no default	The unique identifier.
objects	no default	A list of ids. All units/items in an army which were created from tags with one of the listed ids are counted and their numbers are added. This total has to be within (including) the limit give in the min and max attributes.
min	“0”	The number of all units/items must be equal or exceed this number.
max	“99”	The number of all units/items must not exceed this number.
limit	perK	perK or perA. This defines wether the min and max values are per 1000 points or per army.

The body text will in a later stage be displayed when this error is encountered.

WM:CompLimit

Each WM:CompLimit tag describes one Test which has to be passed for the army list to be correct.

Name	Default	
id	no default	The unique identifier.
left	no default	A list of ids. The count of all units/items in an army which were created from tags with one of the listed ids is taken.
leftPercentage	“100”	The count of all left objects is multiplied by this value and divided by 100. The result is the left value.
right	no default	A list of ids. The count of all units/items in an army which were created from tags with one of the listed ids is taken.

jWM Army File format

Name	Default	
rightPercentage	"100"	The count of all right objects is multiplied by this value and divided by 100. The result is the right value.
relation	no default	One of the following values: less or lessOrEqual or equal or moreOrEqual or more. The test is passed if the left value is in the defined relation to the right value.
or	noOr	One of the following values: all, nothing or noOr. This is a second way to pass the test. If the value is noOr the test is only passed if the relation test is passed. If the value is nothing the test is passed if the relation test is passed or the left value is 0. If the value is all the test is passed if the relation test is passed or the left value is equal to the right value.

The body text is displayed in the error dialog when this error is encountered.

jWM Army File format

Samples and concepts

The following is the blueprint for a race definition:

<code><?xml version="1.0" encoding="UTF-8"?></code>	XML header and DTD reference
<code><!DOCTYPE WM:Race SYSTEM "Race.dtd"></code>	
<code><WM:Race id="R_TEST" name="Test race" armyName="Test army"></code> <code></WM:Race></code>	Race tag

Let's add an entry for the General and a Leader

<code><?xml version="1.0" encoding="UTF-8"?></code>	XML header and DTD reference
<code><!DOCTYPE WM:Race SYSTEM "Race.dtd"></code>	
<code><WM:Race id="R_TEST" name="Test race" armyName="Test army"></code>	Race tag
<code> <WM:Character id="C_GENR" name="General" type="General" atk="+2" cmd="9" pts="125" ></code>	Character Tag describing General
<code> </WM:Character></code>	
<code> <WM:Character id="C_LEAD" name="Leader" type="Leader" atk="+1" cmd="8" pts="80"></code>	Character Tag describing Leader
<code> </WM:Character></code>	
<code></WM:Race></code>	

The General get's an option for Portents and a tag describing the rules for the Portents.

<code><?xml version="1.0" encoding="UTF-8"?></code>	XML header and DTD reference
<code><!DOCTYPE WM:Race SYSTEM "Race.dtd"></code>	
<code><WM:Race id="R_TEST" name="Test race" armyName="Test army"></code>	Race tag
<code> <WM:Character id="C_GENR" name="General" type="General" atk="+2" cmd="9" pts="125" options = "O_PORT" ></code>	Character Tag describing General
<code> </WM:Character></code>	
<code> <WM:Character id="C_LEAD" name="Leader" type="Leader" atk="+1" cmd="8" pts="80"></code>	Character Tag describing Leader
<code> </WM:Character></code>	
<code> <WM:Option id="O_PORT" name="Portents" pts="10" rules="R_PORT"/></code>	Option tag
<code> <WM:Rule id="R_PORT" name="Portents"></code> <code> Portents are the best thing since sliced bread.</code>	Rule Tag
<code> </WM:Rule></code>	
<code></WM:Race></code>	

jWM Army File format

The General get's an additional option for three different mounts. The leader gets an chariot mount option only.

<pre><?xml version="1.0" encoding="UTF-8"?></pre>	XML header and DTD reference
<pre><!DOCTYPE WM:Race SYSTEM "Race.dtd"></pre>	
<pre><WM:Race id="R_TEST" name="Test race" armyName="Test army"></pre>	Race tag
<pre> <WM:Character id="C_GENR" name="General" type="General" atk="+2" cmd="9" pts="125" options = "O_MOUNT O_PORT" ></pre>	The General has now 2 options. The portent option will use a checkbox and the mount option will use a combobox.
<pre></WM:Character></pre>	
<pre> <WM:Character id="C_LEAD" name="Leader" type="Leader" atk="+1" cmd="8" pts="80" options="M_CHAR"></pre>	The Leaders's chariot option will be represented by a checkbox.
<pre></WM:Character></pre>	
<pre> <WM:CharacterUpgrade id="M_CHAR" name="Chariot" type="Mount" atk="+1" pts="10" siz="1" cmd="" /></pre>	Note that this tag is referenced by the Leader tag and the Mount option tag. This is possible because the chariot is identical in all attributes for Generals and Leaders
<pre> <WM:CharacterUpgrade id="M_ELE" name="Elephant" type="Mount" atk="+2" pts="40" siz="1" cmd="" rules="R_ELE"/></pre>	The elephant has it's own special rules
<pre> <WM:CharacterUpgrade id="M_NOMO" name="No Mount" type="Mount" rng="" atk="" pts="0" siz="" cmd="" hide="yes" /></pre>	A no mount is given, so that the user can select a no mount option. When no mount is selected no profile should be displayed. Therefore the hide attribute is set to yes.
<pre><WM:Option id="O_PORT" name="Portents" pts="10" rules="R_PORT"/></pre>	
<pre><WM:ExcOption id="O_MOUNT" name="Mount" options="M_NOMO M_CHAR M_ELE"/></pre>	The Mount option offers three choices. The first choice is the default choice.
<pre><WM:Rule id="R_PORT" name="Portents"> Portents are the best thing since sliced bread. </WM:Rule></pre>	
<pre><WM:Rule id="R_ELE" name="Elephant"/></pre>	Rules for Elephant
<pre></WM:Race></pre>	

jWM Army File format

The General can be upgraded to Alexander with a higher cmd. But Alexander can only ride in a chariot, not on an Elephant.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WM:Race SYSTEM "Race.dtd">
<WM:Race id="R_TEST" name="Test race" armyName="Test army">
  <WM:Character id="C_GENR" name="General" type="General"
    atk="+2" cmd="9" pts="125" upgrades = "C_ALEX" options =
    "O_MOUNT O_PORT ">
  </WM:Character>
  <WM:CharacterUpgrade id="C_ALEX" name="Alexander"
    type="General" atk="+2" cmd="10" pts="135" options =
    "O_CHAR O_PORT "/>

  <WM:Character id="C_LEAD" name="Leader" type="Leader"
    atk="+1" cmd="8" pts="80" options="M_CHAR">
  </WM:Character>
  <WM:CharacterUpgrade id="M_CHAR" name="Chariot"
    type="Mount" atk="+1" pts="10" siz="1" cmd="" />
  <WM:CharacterUpgrade id="M_ELE" name="Elephant"
    type="Mount" atk="+2" pts="40" siz="1" cmd=""
    rules="R_ELE"/>
  <WM:CharacterUpgrade id="M_NOMO" name="No Mount"
    type="Mount" rng="" atk="" pts="0" siz="" cmd="" hide="yes"
    />
  <WM:Option id="O_PORT" name="Portents" pts="10"
    rules="R_PORT"/>
  <WM:ExcOption id="O_MOUNT" name="Mount" op-
    tions="M_NOMO M_CHAR M_ELE"/>
  <WM:Rule id="R_PORT" name="Portents">
    Portents are the best thing since sliced bread.
  </WM:Rule>
  <WM:Rule id="R_ELE" name="Elephant"/>
</WM:Race>
```

The upgrades attribute has been added.

Note that Alexander has no upgrade option and that all attributes have to be set as if Alexander would be an ordinary character not an upgrade.

jWM Army File format

We add the Limits. Alexander or General must exist once per Army. The Leader twice per 1000 points. The chariot option only once per 1000 points,

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WM:Race SYSTEM "Race.dtd">
<WM:Race id="R_TEST" name="Test race" armyName="Test army">
  <WM:Character id="C_GENR" name="General" type="General"
    atk="+2" cmd="9" pts="125" upgrades = "C_ALEX" options =
    "O_MOUNT O_PORT ">
  </WM:Character>
  <WM:CharacterUpgrade id="C_ALEX" name="Alexander"
    type="General" atk="+2" cmd="10" pts="135" options =
    "O_CHAR O_PORT "/>
  <WM:Character id="C_LEAD" name="Leader" type="Leader"
    atk="+1" cmd="8" pts="80" options="M_CHAR">
  </WM:Character>
  <WM:CharacterUpgrade id="M_CHAR" name="Chariot"
    type="Mount" atk="+1" pts="10" siz="1" cmd="" />
  <WM:CharacterUpgrade id="M_ELE" name="Elephant"
    type="Mount" atk="+2" pts="40" siz="1" cmd=""
    rules="R_ELE"/>
  <WM:CharacterUpgrade id="M_NOMO" name="No Mount"
    type="Mount" rng="" atk="" pts="0" siz="" cmd="" hide="yes"
    />
  <WM:Option id="O_PORT" name="Portents" pts="10"
    rules="R_PORT"/>
  <WM:ExcOption id="O_MOUNT" name="Mount" op-
    tions="M_NOMO M_CHAR M_ELE"/>
  <WM:Rule id="R_PORT" name="Portents">
    Portents are the best thing since sliced bread.
  </WM:Rule>
  <WM:Rule id="R_ELE" name="Elephant"/>
  <WM:Limit id="L_GENERAL" objects="C_GENR C_ALEX"
    min="1" max="1" limit="perA" >Number of Generals is
    wrong.</WM:Limit>
  <WM:Limit id="L_LEADER" objects="C_LEAD" min="0"
    max="2" >Too many leaders.</WM:Limit>
  <WM:Limit id="L_CHAR" objects="C_CHAR" min="0" max="1"
    >To many chariot mounts.</WM:Limit>
</WM:Race>
```

jWM Army File format

You should be familiar with the structure of the file by now, so I will only show the relevant tags.

Let's look about a more complicated limit.

The army has a Phalanx. There must be between 2 and 4 Guard Phalanxes per 1000 points. And one Phalanx in the army can be upgraded to Hypaspists. And none or exactly half of the Phalanxes (including Hypaspists) can be upgraded to Imitation Romans.

<pre><WM:Unit id="U_GRPH" name="Guard Phalanx" type="Infantry" atk="3" arm="5+" pts="70" rules="R_PHLX" upgrades="U_HYP U_IR"/></pre>	Phalanx
<pre><WM:UnitUpgrade id="U_HYP" name="Hypaspists" type="Infantry" atk="4" arm="5+" pts="80" rules="R_PHLX"/ ></pre>	Hypaspists
<pre><WM:UnitUpgrade id="U_IR" name="Imitation Romans" type="Infantry" atk="3" arm="5+" pts="65" rules="R_MANI"/></pre>	Imitation Romans
<pre><WM:Rule id="R_MANI" name="Manipel"> A Manipel is a Manipel is a Manipel.</WM:Rule></pre>	The rules
<pre><WM:Rule id="R_PHLX" name="Phalanx"/></pre>	
<pre><WM:Limit id="L_GRPH" objects="U_GRPH U_HYP U_IR" min="2" max="4"> Number of Guard Phalanxes and derivatives is wrong. </WM:Limit></pre>	2 to 4 units in any combination of those types
<pre><WM:Limit id="L_HYP" objects="U_HYP" min="0" max="1" limit="perA" > Too many Hypaspists. </WM:Limit></pre>	
<pre><WM:CompLimit id="L_IR" left = "U_IR" relation = "equal" right = "U_GRPH U_HYP U_IR" rightPercentage = "50" or = "nothing" > Number of Imitation Romans is wrong. </WM:CompLimit></pre>	The number of U_IR must be equal to 50% of the sum of all U_IR + U_HYP + U_GRPH or U_IR must be 0.

jWM Army File format

Instead of using upgrades the different types of Phalanxes could have been defined as different units instead of upgrades. As long as the limits stay as they are the „logic“ stays the same, but the presentation in jWM is different. Try it out when you have the time.

<WM:Unit id="U_GRPH" name="Guard Phalanx" type="Infantry" atk="3" arm="5+" pts="70" rules="R_PHLX"/>	Phalanx
<WM:Unit id="U_HYP" name="Hypaspists" type="Infantry" atk="4" arm="5+" pts="80" rules="R_PHLX"/ >	Hypaspists
<WM:Unit id="U_IR" name="Imitation Romans" type="Infantry" atk="3" arm="5+" pts="65" rules="R_MANI"/>	Imitation Romans
<WM:Rule id="R_MANI" name="Manipel"> A Manipel is a Manipel is a Manipel.</WM:Rule>	The rules
<WM:Rule id="R_PHLX" name="Phalanx"/>	
<WM:Limit id="L_GRPH" objects="U_GRPH U_HYP U_IR" min="2" max="4"> Number of Guard Phalanxes and derivatives is wrong.	2 to 4 units in any combination of those types
</WM:Limit>	
<WM:Limit id="L_HYP" objects="U_HYP" min="0" max="1" limit="perA" > Too many Hypaspists.	
</WM:Limit>	
<WM:CompLimit id="L_IR" left = "U_IR" relation = "equal" right = "U_GRPH U_HYP U_IR" rightPercentage = "50" or = "nothing" > Number of Imitation Romans is wrong.	The number of U_IR must be equal to 50% of the sum of all U_IR + U_HYP + U_GRPH or U_IR must be 0.
</WM:CompLimit>	