

## Armory 201 – Conflict Checking Using the Complex Search Form

Presented by  
Yehuda ben Moshe  
Elmet Herald

1

---

---

---

---

---

---

---

---

## Prerequisites

- A thorough understanding of the conflict rules and how to conflict check with the Ordinary Index
- Students are encouraged to review Armory 103 and 104, as well as SENA A5 prior to this class

2

---

---

---

---

---

---

---

---

## Review

- Remember to always look for SCs first, before counting DCs!
- Any **one** SC will clear the device
  - Type
  - Number
  - Arrangement
  - Posture/Orientation
  - (Adding/removing a primary charge group)

3

---

---

---

---

---

---

---

---

### Review

- Any **two** DCs will clear the device
  - Field
  - Adding/removing a charge group
  - Tincture within a charge group
  - Type
  - Number
  - Arrangement
  - Posture/Orientation

4

---

---

---

---

---

---

---

---

### Where

- All SCA-registered armory (and names) are located in the database known as the Ordinary and Armorial
- [oanda.sca.org](http://oanda.sca.org)
- Different ways to search – complex search form

5

---

---

---

---

---

---

---

---

### Warning

- Do **NOT** use the blazon pattern search for conflict checking. You **WILL** miss conflicts that way
  - Simple text search
  - Consider conflict checking a “dog”
  - Need to also check for “talbot”, “wolf”, “greyhound”, “fox”, etc.
- Only use Ordinary index and Complex Search form for conflict checking

6

---

---

---

---

---

---

---

---

### Background

- Complex Search Form is actually just a way to run multiple searches at once
- While you can do different types of searches, we will be using only the Armory Description type of search
- When properly used, the complex search form will give the exact same results as the Ordinary Index – you'll get all actual conflicts and eliminate all non-conflicts
  - But the Complex Form is faster – less to hand check

7

---

---

---

---

---

---

---

---

### Background

- Every entry in the O and A is coded using a system of armory descriptions
- An armory description is comprised of a category and zero or more features
- Example: CAT:3:primary:rampant to dexter:or
  - CAT is the category
  - 3, primary, rampant to dexter, or are all features
- Categories are all-capital letters, features are all lower case

8

---

---

---

---

---

---

---

---

### Background

- NOTE: not all entries in the O and A are fully coded
  - All have categories, but not all features are fully entered
  - More recent registrations are more likely to be fully coded
  - Older registrations are being re-coded by Morsulus Herald
  - Items that are not fully coded will show up unexpectedly, but you will not miss a conflict because of them

9

---

---

---

---

---

---

---

---

## Background

- To see examples of coding, perform a Blazon Pattern Search and set "Armory descriptions" to "enabled"
  - Remember: Don't conflict check with the Blazon Pattern Search

10

---

---

---

---

---

---

---

---

## my.cat

- All codes are defined in a file located on the O and A site:
  - <http://oanda.sca.org/my.cat>
  - Some computers have trouble displaying this file – if so, save it and rename it as a .txt file, then open in a text editor
- This is an automatically generated file that shows all categories and features, as well as their relationships to each other
- It does not, however, show how they are used

11

---

---

---

---

---

---

---

---

## my.cat

- The file has three main sections
  - Features
  - Categories
  - Cross-references
- We will look at them in reverse order

12

---

---

---

---

---

---

---

---

## my.cat

- Cross-references
  - The bottom part of the file contains all of the cross-references
  - These are the same as the “see X” entries in the Ordinary Index
  - One way to find codes is to do a search in the file (Ctrl-F on most PCs) and look for your text

13

---

---

---

---

---

---

---

---

## my.cat

- Cross-references
  - Example: suppose we have a device with a yak on it.
    - Perform a search for “yak” and we will find the following: “yak – see beast, bull”
    - This is a cross-reference
    - We should now search for “beast, bull” to find the actual category

14

---

---

---

---

---

---

---

---

## my.cat

- Categories
  - Categories show how the Armory Description is coded
  - Remember categories are always capitalized
  - Example: continuing from above
    - Searching for “beast, bull” finds the following line:
    - “beast, bull|BEAST-BULL AND BISON”
    - This tells us that “BEAST-BULL AND BISON” is the coding for a Yak (and other bull-type creatures)

15

---

---

---

---

---

---

---

---

## my.cat

- Categories
  - A category will always have a lower case “plain English” term, followed by a pipe symbol (“|”) followed by the all-capital category code

16

---

---

---

---

---

---

---

---

## my.cat

- Features
  - Features are organized into feature groups
    - Feature groups don’t appear anywhere other than in my.cat
    - They are used internally by the system
    - Features are compared only to other features in the same group
      - For example, the features in the feature group “posture”, such as rampant, statant, etc. will only match other features in the “posture” group

17

---

---

---

---

---

---

---

---

## my.cat

- Features
  - Features always start with a pipe character, then the feature group, then a colon, then the feature
    - |posture:passant
  - Features can also be included in other features
    - |posture:passant to dexter<passant
    - This means that “passant” will also match “passant to dexter”

18

---

---

---

---

---

---

---

---

## my.cat

- Features
  - These are not subsets
    - |number:4<4 or fewer<4 or more<2 or more
    - This means that “2 or more”, “4 or more”, “4 or fewer” will all match “4”
    - It does not mean that “4 or more” will match “2 or more”
  - They only describe what will match to the left most item

19

---

---

---

---

---

---

---

---

## Matching

- The complex search form tries to match each line of the search to the description of the registered armory
- The system doesn't actually look for a match – it looks for a mismatch
  - This is very important to understand

20

---

---

---

---

---

---

---

---

## Matching

- Consider “Gules, a lion Or” as a registered item
- The lion in this entry might be encoded as:
  - CAT:or:1:spa:rampant to dexter
    - CAT is the category
    - “or” is a feature in the “tincture” group
    - “1” is a feature in the “number” group
    - “spa” (single primary alone on the field) is a feature in the “group” group
    - “rampant to dexter” is a feature in the “posture” group

21

---

---

---

---

---

---

---

---

### Matching

- Suppose we are trying to conflict check “Vert, a lion argent”
- We might try the following searches
  - CAT:1
    - The system first checks that the category is present
    - It then notices that the feature is “1” and that that feature is in the “number” group
    - Next, it checks that the registered item has a feature in the “number” group – which it does
    - Finally, it checks if the features match – they do, so this search is a match

22

---

---

---

---

---

---

---

---

### Matching

- Suppose we are trying to conflict check “Vert, a lion argent”
- We might try the following searches
  - CAT:argent
    - The system first checks that the category is present
    - It then notices that the feature is “argent” and that that feature is in the “tincture” group
    - Next, it checks that the registered item has a feature in the “tincture” group – which it does
    - Finally, it checks if the features match – they don’t, so it’s a mismatch

23

---

---

---

---

---

---

---

---

### Matching

- Suppose we are trying to conflict check “Vert, a lion argent”
- Suppose we make an error, and misspell argent
  - CAT:arget
    - The system first checks that the category is present
    - It then notices that the feature is “arget” but that this doesn’t belong to any feature group
    - Since it is not part of a feature group, it is ignored
    - Since the system looks for mismatches, not matches, this returns a match

24

---

---

---

---

---

---

---

---

### Matching

- Suppose we are trying to conflict check “Vert, a lion argent”
- Suppose we make an error, and use a totally wrong (but valid) feature
  - CAT:naiant
    - The system first checks that the category is present
    - It then notices that the feature is “naiant”, which belongs to the “fish\_posture” group
    - Next, it checks that the registered item has a feature in the “fish\_posture” group – but it doesn’t
    - That group isn’t present, so this feature is ignored.
    - Since the system looks for mismatches, not matches, this returns a match

25

---

---

---

---

---

---

---

---

### Matching

- Suppose we are trying to conflict check “Vert, a lion argent”
- Now, suppose the registered item is old, and not fully indexed. Instead of the full indexing we had above, it’s just indexed as
  - CAT
- We try to search for “CAT:argent”, a correct search
  - The system first checks that the category is present
  - It then notices that the feature is “argent”, which belongs to the “tincture” group
  - Next, it checks that the registered item has a feature in the “tincture” group – but it doesn’t
  - That group isn’t present, so this feature is ignored.
  - Since the system looks for mismatches, not matches, this returns a match

26

---

---

---

---

---

---

---

---

### Matching

- What this means is that we get a mismatch **only** when the system is sure there is no match
  - Otherwise, the system returns a match

27

---

---

---

---

---

---

---

---

### Finding Codes

- The category and feature codes are voluminous, and impossible to memorize
- They can be found in my.cat
  - Ctrl-F is your friend
- They can be found using the blazon pattern search form
  - Set “Armory descriptions” to “enabled”
  - Search for what you need and read the coding
  - REMEMBER: Do not conflict search with this form

28

---

---

---

---

---

---

---

---

### Using the form

- The complex form has 10 lines into which searches can be entered
- Each line can be assigned a “weight” and can be assigned the type of search
  - By default, all lines get a weight of “1” and an “armory description” search type
  - These defaults are exactly what we want, do not change them unless you know what you are doing
- Any line where nothing is entered into the “pattern” box is ignored completely

29

---

---

---

---

---

---

---

---

### Using the form

- When the search is run, each line is assigned a value, equal to the “weight” of that line if the search matches, or 0 if the search doesn’t match
- Therefore, if we use five lines, and all five lines match, the score will be 5. If three lines match, the score will be 3.

30

---

---

---

---

---

---

---

---

### Using the form

- The goal is to have each line be worth exactly one DC
  - That way, if the line doesn't match, it means we have one DC
  - If the line matches, it means either that that DC isn't present, or that the system couldn't tell for sure

31

---

---

---

---

---

---

---

---

### Using the form

- In this way, if the maximum score (number of lines) is n, we can ignore any results with a score of n-2 or less – they are guaranteed to have at least 2 DCs
- Any result with a score of n or n-1 needs to be hand-checked
  - These might be conflicts
  - They might be clear but not properly indexed
  - They might have DCs for things not easily coded, such as arrangement

32

---

---

---

---

---

---

---

---

### Constructing Search

- Identify all charge groups
  - Including all details that are worth DCs
- Use one line to code field
  - Since only one DC is possible from the field, we want to use only one line
- Use as many lines as possible to code primary group
  - We want to generate as many DCs as possible

33

---

---

---

---

---

---

---

---

## Constructing Search

- Use one line for each of the other charge group
  - Multiple DCs are possible, so it is tempting to use multiple lines
  - However, removing a charge group entirely is worth only one DC, so if we use multiple lines and the group is removed, we'd get too many DCs

34

---

---

---

---

---

---

---

---

## Sample

- Blazon: Per pale azure and gules, a lion and on a chief indented Or, three escallops gules.
  - Field: Per pale azure and gules, plain line
  - Primary charge group: lion, 1, Or, rampant
  - Secondary charge group: chief, 1, Or, indented line
  - Tertiary charge group: escallop, 3, gules, in fess




---

---

---

---

---

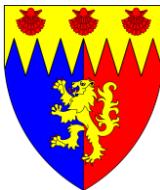
---

---

---

## Sample

- Search
  - PPALE:azure:~and gules:pl
  - CAT:1:primary
  - CAT:or:primary
  - CAT:rampant to dexter:primary
  - CHIEF:or:jagged:1:secondary
  - ESCALLOP:3:gules:tertiary




---

---

---

---

---

---

---

---

## Practice

- Blazon: Per bend argent and Or, a tower and a chief gules.
- Field: per bend argent and Or, plain line
- Primary charge group: Tower, 1, gules
- Secondary charge group: Chief, 1, gules, plain line




---

---

---

---

---

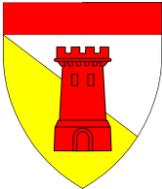
---

---

---

## Practice

- Search:
  - PBEND:argent::~and Or:pl
  - CASTLE:1:primary
  - CASTLE:gules:primary
  - CHIEF:gules:pl:unc




---

---

---

---

---

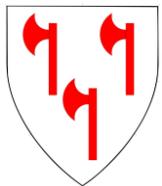
---

---

---

## Practice

- Blazon: Argent, three axes gules
- Field: Argent
- Primary charge group: Axe, 3, gules, palewise, to chief, 2 and 1
- No other charges




---

---

---

---

---

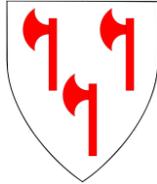
---

---

---

## Practice

- Search:
  - AR
  - AXE:3:g3pa
  - AXE:gules:primary




---

---

---

---

---

---

---

---

## Split-Type Charge Groups

- When a charge group has two types, they must be checked separately
- Trying to check in one search will miss conflicts
- (Note: There is an advanced technique that can be used to do this in one search, but it is not covered in this class)

41

---

---

---

---

---

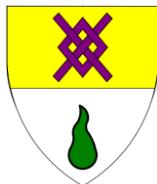
---

---

---

## Practice

- Blazon: Per fess Or and argent, a fret couped purpure and a goutte vert. (or goutte d'huile)
- Field: Per fess Or and argent, plain line
- Primary charge group: Fret, 1, purpure AND Goutte, 1, vert




---

---

---

---

---

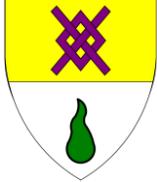
---

---

---

### Practice

- Incorrect search:
  - PFESS:pl:or::~and argent
  - FRET:1:primary
  - FRET:purple:primary
  - GOUTE:1:primary
  - GOUTE:vert:primary
- Consider: "Per fess Or and argent, a fret purple and a fret vert"
- What would the score be?



---

---

---

---

---

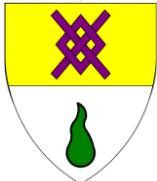
---

---

---

### Practice

- Correct search 1:
  - PFESS:pl:or::~and argent
  - FRET:g2pa
  - FRET:purple:primary



---

---

---

---

---

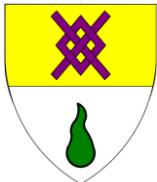
---

---

---

### Practice

- Correct search 2:
  - PFESS:pl:or::~and argent
  - GOUTE:g2pa
  - GOUTE:vert:primary



---

---

---

---

---

---

---

---

## Fieldless Badges

- When something is fieldless, it uses the “NO” code for the field
- Do not use this code
- Simply skip the field, and only check things that have a maximum score – anything with a score of n-1 has one DC from search and the fieldless DC

46

---

---

---

---

---

---

---

---

## Field Primary Armory

- I don't know of any good way to conflict check field primary armory this way
- I just go to the Ordinary and check the “Field only” and “Peripheral only” categories by hand

47

---

---

---

---

---

---

---

---

## Pitfalls

- There is a pair of features “charged” (cha) and “uncharged” (unc).
  - Uncharged is very useful
  - Charged should be avoided
    - Consider “Or, a lion sable, on a chief gules a sword argent”
    - We can code it as:
      - OR
      - CHIEF:gules:cha
      - SWORD:argent:1:tertiary
    - What if there is a registered “Or, a lion sable, a chief gules” – how many DCs will the search show?
    - (I will use “charged” when the tertiaries are split-type, and not code the tertiary group)

48

---

---

---

---

---

---

---

---

## Pitfalls

- Similarly, primary charges can be alone (spa, gpa, g2pa, etc.) and not alone (spna, gpna, g4pna, etc.).
  - Alone is very useful
  - “Not alone” should be avoided
    - Consider “Or, a lion sable between three swords gules”
    - We can code it as:
      - OR
      - CAT:1:spna
      - SWORD:gules:3:secondary
    - What if there is a registered “Or, a lion sable” – how many DCs will the search show?
    - (Again, I’ll use “not alone” with split-type secondary groups and not code the secondary)

49

---

---

---

---

---

---

---

---

## Summary

- Remember – one line for field, as many as possible for primary, one per group for other groups
- For mixed groups, check both
- Avoid uncharged and not alone

50

---

---

---

---

---

---

---

---

## Resources

- Article on using the complex form can be found here:
  - <http://www.tanzos.net/~victoria/HeraldicEducation/>

51

---

---

---

---

---

---

---

---

### Final Thoughts

- Registerable vs. Authentic
  - A submission must be registerable; it need not be authentic
  - While we can encourage clients to design period-looking armory, we cannot, and should not, force the decision
  - If a client is set on a registerable but not very authentic submission, you should process it!

52

---

---

---

---

---

---

---

---

### Final Thoughts

- Customer Service
  - It is our job to help our clients
  - We are here to make registrations happen, not prevent them from happening
  - When consulting, help clients create registerable submissions **they** like
  - When commenting, look for reasons to allow registration, not prohibit it
  - Heralds want a reputation for being helpful, not obstructionist!

53

---

---

---

---

---

---

---

---

### About me

- Elmet Herald – I am the East Kingdom heraldic education deputy
- [elmet@eastkingdom.org](mailto:elmet@eastkingdom.org)
- [jgalak@gmail.com](mailto:jgalak@gmail.com)
- This handout can be found at:
  - <http://elmet.eastkingdom.org>

54

---

---

---

---

---

---

---

---