

# zCapacity Events — XML Event Documentation

This documentation explains how to create XML events for **zCapacity Events**, including how to correctly use selected characters, relationships, pregnancy logic, prisoner lords, family members, lords, companions, notables and army-related interactions.

If you want to create your own events, it is strongly recommended to first inspect the included example files:

```
Modules\zCaptivityEvents\ModuleLoader\Events-zMojeCE  
  
ZC_ArmyLords_Menu.xml  
ZC_Companion_Menu.xml  
ZC_Family_Menu.xml  
ZC_PrisonerLord_Menu.xml  
ZC_ServiceArmy_Menu_auto.xml  
ZC_SettlementLord_Menu.xml  
zMojeCE_Burdel.xml
```

These files demonstrate practical usage of the zCapacity system in working XML events.

---

## 1. Basic Event File Structure

Every event file must use this root element:

```
<CEEEvents>  
  <CEEEvent>  
    ...  
  </CEEEvent>  
</CEEEvents>
```

Do **not** use:

```
<Events>
```

That root will not load correctly.

Minimal event example:

```
<?xml version="1.0" encoding="UTF-8"?>  
<CEEEvents>  
  <CEEEvent>  
    <Name>ZC_My_Test_Event</Name>
```

```

<Text>This is a test event.</Text>
<BackgroundName>CE_common_settlement</BackgroundName>

<MultipleRestrictedListOfFlags>
  <RestrictedListOfFlags>Random</RestrictedListOfFlags>
  <RestrictedListOfFlags>CanOnlyBeTriggeredByOtherEvent</
RestrictedListOfFlags>
  <RestrictedListOfFlags>Common</RestrictedListOfFlags>
</MultipleRestrictedListOfFlags>

<Options>
  <Option>
    <Order>0</Order>
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>Leave</RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
    <OptionText>Leave.</OptionText>
  </Option>
</Options>

<ReqCustomCode>>true</ReqCustomCode>
<SexualContent>>false</SexualContent>
</CEEEvent>
</CEEEvents>

```

Important fields:

Name	- unique event name
Text	- event text
BackgroundName	- background image name without .png
MultipleRestrictedListOfFlags	- required CE flags
Options	- player choices
ReqCustomCode	- usually true for zCapacity events
SexualContent	- true/false

## 2. zCapacity Menus

zCapacity uses special menu router events. Each router has an exact XML name. The DLL looks for these names directly, so a typo will prevent the menu from opening.

Main zCapacity menu names:

ZC_Service_Companions_Menu	- companion interactions
ZC_Service_Family_Menu	- family member interactions
ZC_Service_PrisonerLords_Menu	- prisoner lord interactions
ZC_Service_SettlementLord_Menu	- settlement lord/lady interactions

ZC_Service_ArmyLords_Menu	- army lord interactions
ZC_Service_Notables_Menu	- notable interactions in towns/villages
ZC_Service_Soldiers_Menu	- own troop interactions
ZC_Service_ArmySoldiers_Menu	- army troop interactions

Only one active XML event should use a given menu name.

Bad:

```
Active file A contains ZC_Service_Family_Menu
Active file B also contains ZC_Service_Family_Menu
```

Good:

```
Only one active XML file contains ZC_Service_Family_Menu
```

When testing alternative versions, disable old files by renaming them to:

```
.xml.disabled
```

---

## 3. Menu Router Example

A menu router is an event that displays options and sends the player to other events.

Example:

```
<CEEvent>
  <Name>ZC_Service_Family_Menu</Name>
  <Text>Choose what you want to do with the selected family member.</Text>
  <BackgroundName>CE_common_settlement</BackgroundName>

  <MultipleRestrictedListOfFlags>
    <RestrictedListOfFlags>Random</RestrictedListOfFlags>
    <RestrictedListOfFlags>CanOnlyBeTriggeredByOtherEvent</
RestrictedListOfFlags>
    <RestrictedListOfFlags>Common</RestrictedListOfFlags>
  </MultipleRestrictedListOfFlags>

  <Options>
    <Option>
      <Order>0</Order>
      <MultipleRestrictedListOfConsequences>
        <RestrictedListOfConsequences>Continue</RestrictedListOfConsequences>
        <RestrictedListOfConsequences>EmptyIcon</
```

```

RestrictedListOfConsequences>
  </MultipleRestrictedListOfConsequences>
  <OptionText>Talk privately.</OptionText>
  <TriggerEventName>ZC_Family_Private_Talk_01</TriggerEventName>
</Option>

<Option>
  <Order>99</Order>
  <MultipleRestrictedListOfConsequences>
    <RestrictedListOfConsequences>Leave</RestrictedListOfConsequences>
  </MultipleRestrictedListOfConsequences>
  <OptionText>Leave.</OptionText>
</Option>
</Options>

<ReqCustomCode>>true</ReqCustomCode>
<SexualContent>>false</SexualContent>
</CEEEvent>

```

Important:

TriggerEventName starts another event by its exact name.

## 4. Hero References

zCapacity adds a reference system. These references allow XML events to operate on the exact character selected by the player.

Main references:

SelectedActor	- currently selected acting hero
SelectedFamilyMember	- selected family member
SelectedCompanion	- selected companion
SelectedPrisonerLord	- selected prisoner lord
SelectedLord	- selected lord/lady from settlement or army
SelectedSettlementLord	- alias for settlement lord
SelectedNotable	- selected notable
SelectedHero	- general selected target
SelectedTarget	- general selected target
MainHero	- player character
Player	- player character

Safest references for normal events:

```
SelectedActor  
SelectedFamilyMember  
SelectedCompanion  
SelectedPrisonerLord  
SelectedLord  
SelectedNotable
```

Avoid using `Hero` and `EventHero` when you want to target a specific selected character. These can resolve to an unexpected target.

---

## 5. SelectedActor — Who Performs the Interaction

`SelectedActor` is the character performing the interaction.

By default, the actor is usually `MainHero`, but `zCapacity` allows the player to change the active actor. This makes it possible to create events such as:

```
brother -> sister  
companion -> lord  
family member -> notable  
player -> prisoner lord
```

Most commonly used XML pattern:

```
Ref="SelectedFamilyMember"  
RelationRef="SelectedActor"
```

This means:

```
The main target is SelectedFamilyMember.  
The other side of the relationship is SelectedActor.
```

---

## 6. Effects on a Specific Hero — Companion

Most effects that operate on a specific hero should be placed inside:

```
<Companions>  
  <Companion Ref="..." Type="...">  
    ...
```

```
</Companion>
</Companions>
```

Example:

```
<Companions>
  <Companion Ref="SelectedCompanion" Type="SelectedCompanion">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ChangeRelation</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
    <RelationTotal>5</RelationTotal>
  </Companion>
</Companions>
```

Do not place effects that require a specific hero directly at the option level unless the DLL does not need to resolve a target.

Bad:

```
<MultipleRestrictedListOfConsequences>
  <RestrictedListOfConsequences>ZC_KillHeroByRef</
RestrictedListOfConsequences>
</MultipleRestrictedListOfConsequences>
```

Good:

```
<Companions>
  <Companion Ref="SelectedPrisonerLord" Type="SelectedPrisonerLord">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ZC_KillHeroByRef</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
  </Companion>
</Companions>
```

---

## 7. Relationships Between Selected Heroes

Use:

```
RelationRef="SelectedActor"
```

Example: relationship change between the selected family member and the current actor:

```
<Companions>
  <Companion Ref="SelectedFamilyMember"
            Type="SelectedFamilyMember"
            RelationRef="SelectedActor">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ChangeRelation</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
    <RelationTotal>5</RelationTotal>
  </Companion>
</Companions>
```

Result:

SelectedFamilyMember receives +5 relationship with SelectedActor.

Negative relationship example:

```
<Companions>
  <Companion Ref="SelectedLord"
            Type="SelectedLord"
            RelationRef="SelectedActor">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ChangeRelation</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
    <RelationTotal>-10</RelationTotal>
  </Companion>
</Companions>
```

---

## 8. Reference-Based Pregnancy — ImpregnationByRef

Use:

ImpregnationByRef

for pregnancy logic between two specific selected heroes.

Example:

```
<Companions>
  <Companion Ref="SelectedFamilyMember"
    Type="SelectedFamilyMember"
    PregnancyTargetRef="SelectedFamilyMember"
    PregnancyPartnerRef="SelectedActor"
    AdultOnly="true"
    AllowPregnant="false"
    PregnancyForce="true">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ImpregnationByRef</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
    <PregnancyRiskModifier>100</PregnancyRiskModifier>
  </Companion>
</Companions>
```

What the DLL does:

- resolves SelectedFamilyMember
- resolves SelectedActor
- checks gender
- checks adulthood
- checks pregnancy status
- applies the effect only if the pair is valid

The DLL automatically determines direction:

```
female hero = pregnancy target
male hero   = pregnancy partner
```

If both characters are the same sex, underage, invalid or already blocked by pregnancy rules, the effect is skipped.

---

## 9. Forcing Pregnancy

To force the effect, use:

```
PregnancyForce="true"
```

and:

```
<PregnancyRiskModifier>100</PregnancyRiskModifier>
```

Example:

```
<Companion Ref="SelectedCompanion"
  Type="SelectedCompanion"
  PregnancyTargetRef="SelectedCompanion"
  PregnancyPartnerRef="SelectedActor"
  AdultOnly="true"
  AllowPregnant="false"
  PregnancyForce="true">
  <MultipleRestrictedListOfConsequences>
    <RestrictedListOfConsequences>ImpregnationByRef</
RestrictedListOfConsequences>
  </MultipleRestrictedListOfConsequences>
  <PregnancyRiskModifier>100</PregnancyRiskModifier>
</Companion>
```

Recommendation:

Use PregnancyForce="true" only when the result is supposed to be guaranteed. For normal events, use a lower PregnancyRiskModifier.

## 10. Group Pregnancy — ImpregnationByGroupRefs

Use:

```
ImpregnationByGroupRefs
```

for group-based events.

Example for army lords:

```
<Companions>
  <Companion Ref="SelectedLord"
    Type="SelectedLord"
    PregnancyTargetRef="SelectedLord"
    PregnancyPartnerRef="SelectedActor"
    PregnancyGroupType="ArmyLord"
    PregnancyGroupLocation="currentarmy"
    PregnancyGroupCount="8"
    PregnancyMaxTargets="3"
    PregnancyIncludeActor="true"
    PregnancyIncludeSelected="true"
```

```

        AdultOnly="true"
        AllowPregnant="false">
    <MultipleRestrictedListOfConsequences>
        <RestrictedListOfConsequences>ImpregnationByGroupRefs</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
    <PregnancyRiskModifier>50</PregnancyRiskModifier>
</Companion>
</Companions>

```

Important attributes:

PregnancyGroupType	- group type, for example ArmyLord
PregnancyGroupLocation	- where candidates are taken from, for example currentarmy
PregnancyGroupCount	- how many participants to include
PregnancyMaxTargets	- maximum number of targets affected
PregnancyIncludeActor	- whether SelectedActor is added to the group
PregnancyIncludeSelected	- whether the selected target is added to the group

## 11. Killing a Selected Hero — ZC\_KillHeroByRef

Use:

```
ZC_KillHeroByRef
```

to kill a specific selected hero.

Example: killing the selected prisoner lord:

```

<Companions>
  <Companion Ref="SelectedPrisonerLord" Type="SelectedPrisonerLord">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ZC_KillHeroByRef</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
  </Companion>
</Companions>

```

Example: killing the selected family member:

```

<Companions>
  <Companion Ref="SelectedFamilyMember" Type="SelectedFamilyMember">

```

```
<MultipleRestrictedListOfConsequences>
  <RestrictedListOfConsequences>ZC_KillHeroByRef</
RestrictedListOfConsequences>
</MultipleRestrictedListOfConsequences>
</Companion>
</Companions>
```

Important rules:

- always place ZC\_KillHeroByRef inside Companion
- always set Ref
- do not use Hero as Ref
- MainHero is protected from accidental killing by default

---

## 12. Wounding a Selected Hero — ZC\_WoundHeroByRef

Example:

```
<Companions>
  <Companion Ref="SelectedPrisonerLord" Type="SelectedPrisonerLord">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ZC_WoundHeroByRef</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
    <HealthTotal>-50</HealthTotal>
  </Companion>
</Companions>
```

Result:

```
SelectedPrisonerLord loses health according to HealthTotal.
```

---

## 13. Releasing a Prisoner Lord — ZC\_ReleasePrisonerLordByRef

Example:

```

<Companions>
  <Companion Ref="SelectedPrisonerLord" Type="SelectedPrisonerLord">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ZC_ReleasePrisonerLordByRef</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
  </Companion>
</Companions>

```

This effect only works if the selected hero is actually a prisoner in the player's party.

## 14. Custom Skills by Reference

If standard `SkillsToLevel` does not work reliably with selected references, use:

```
ZC_ChangeCustomSkillByRef
```

Example:

```

<Companions>
  <Companion Ref="SelectedActor"
    Type="SelectedActor"
    SkillId="DR_Esteem"
    SkillChange="1"
    SkillMin="0"
    SkillMax="300">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ZC_ChangeCustomSkillByRef</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
  </Companion>
</Companions>

```

Attributes:

SkillId	- skill ID from CEModuleCustom.xml
SkillChange	- value change
SkillSet	- set exact value
SkillMin	- minimum value after change
SkillMax	- maximum value after change

## 15. Standard SkillsToLevel

Example:

```
<SkillsToLevel>
  <Skill Id="DR_Esteem" ByLevel="1" Ref="SelectedActor" />
</SkillsToLevel>
```

Recommendation:

For simple skill changes, SkillsToLevel may be enough.  
For changes on a specific selected character, ZC\_ChangeCustomSkillByRef is safer.

---

## 16. Options and Event Transitions

An option can end the event:

```
<RestrictedListOfConsequences>Leave</RestrictedListOfConsequences>
```

It can continue to another event:

```
<TriggerEventName>ZC_Next_Event</TriggerEventName>
```

It can also randomly choose the next event:

```
<TriggerEvents>
  <TriggerEvent>
    <EventName>ZC_Result_01</EventName>
    <EventWeight>5</EventWeight>
  </TriggerEvent>
  <TriggerEvent>
    <EventName>ZC_Result_02</EventName>
    <EventWeight>1</EventWeight>
  </TriggerEvent>
</TriggerEvents>
```

Example option that starts another event:

```
<Option>
  <Order>0</Order>
```

```
<MultipleRestrictedListOfConsequences>
  <RestrictedListOfConsequences>Continue</RestrictedListOfConsequences>
  <RestrictedListOfConsequences>EmptyIcon</RestrictedListOfConsequences>
</MultipleRestrictedListOfConsequences>
<OptionText>Continue.</OptionText>
<TriggerEventName>ZC_My_Next_Event</TriggerEventName>
</Option>
```

---

## 17. Age Conditions and Safety

For events involving heroes, it is recommended to use:

```
<ReqHeroMinAge>18</ReqHeroMinAge>
```

Inside `Companion`, use:

```
AdultOnly="true"
```

Example:

```
<Companion Ref="SelectedFamilyMember"
  Type="SelectedFamilyMember"
  AdultOnly="true">
```

For pregnancy-related logic, also use:

```
AllowPregnant="false"
```

---

## 18. Brothel and Service Events

Expanded brothel events are located in:

```
zMojeCE_Burdel.xml
```

This file demonstrates:

- how to build longer event trees
- how to create a customer roleplay path
- how to transition between events with `TriggerEventName`

- how to use skills and reputation values
- how to build multi-level menus

If you want to create your own service-style menu, this file is a good structural example.

---

## 19. Army and Troop Events

Army-related service events are located in:

```
ZC_ServiceArmy_Menu_auto.xml
```

The system automatically sets this helper skill:

```
DR_TroopTier_Lvl
```

based on troop count.

This allows XML events to use conditions depending on the size of the party or army.

Example condition:

```
<SkillsRequired>  
  <SkillRequired Id="DR_TroopTier_Lvl" Min="3" Ref="SelectedActor" />  
</SkillsRequired>
```

---

## 20. Common Mistakes

**Event is Found, but not Added**

Check:

```
LoadingFailedXML.txt
```

Common causes:

- wrong root: <Events> instead of <CEEvents>
- typo in RestrictedListOfConsequences
- missing entry in CEEventsModal.xsd

- invalid Companion attribute
- duplicated ZC\_Service\_\*\_Menu

## Effect works on the wrong hero

Most common cause:

```
Hero was used instead of SelectedFamilyMember / SelectedLord / SelectedActor
```

## Killing does not work

Check that the effect is inside:

```
<Companions>  
  <Companion Ref="SelectedPrisonerLord" Type="SelectedPrisonerLord">
```

Adding `ZC_KillHeroByRef` directly to the option is not enough.

## Pregnancy does not work

Check:

- whether both references resolve correctly
- whether one character is female and the other is male
- whether the target is already pregnant
- whether `AdultOnly="true"` is present
- whether `PregnancyRiskModifier` is high enough
- whether `PregnancyForce="true"` is used for testing

---

## 21. Recommended Event Writing Pattern

Safest structure:

1. Menu router selects the category.
2. Option starts a specific event through `TriggerEventName`.
3. Event describes the situation.
4. Final options apply effects through `Companion Ref="..."`
5. Relationship, pregnancy, killing or skill effects are placed inside `Companion`.

Example:

```
<Option>
  <Order>1</Order>
  <MultipleRestrictedListOfConsequences>
    <RestrictedListOfConsequences>Continue</RestrictedListOfConsequences>
    <RestrictedListOfConsequences>EmptyIcon</RestrictedListOfConsequences>
  </MultipleRestrictedListOfConsequences>
  <OptionText>Start private event.</OptionText>
  <TriggerEventName>ZC_Private_Event_01</TriggerEventName>
</Option>
```

Then, in the final event:

```
<Companions>
  <Companion Ref="SelectedFamilyMember"
    Type="SelectedFamilyMember"
    RelationRef="SelectedActor">
    <MultipleRestrictedListOfConsequences>
      <RestrictedListOfConsequences>ChangeRelation</
RestrictedListOfConsequences>
    </MultipleRestrictedListOfConsequences>
    <RelationTotal>5</RelationTotal>
  </Companion>
</Companions>
```

---

## 22. Files Worth Studying

For your own event development, inspect these files:

```
ZC_Family_Menu.xml
```

Shows family events and usage of `SelectedFamilyMember`.

```
ZC_Companion_Menu.xml
```

Shows companion interactions.

```
ZC_PrisonerLord_Menu.xml
```

Shows usage of `SelectedPrisonerLord`, including prisoner lord effects.

```
ZC_SettlementLord_Menu.xml
```

Shows integration with settlement lords and ladies.

```
ZC_ArmyLords_Menu.xml
```

Shows interactions with lords in armies.

```
ZC_ServiceArmy_Menu_auto.xml
```

Shows automatic scaling based on troop count.

```
zMojeCE_Burdel.xml
```

Shows expanded event trees and transitions between scenes.

---

## 23. Minimal Checklist Before Publishing an Event

Before releasing your XML, check:

```
[ ] root is <CEEEvents>
[ ] every CEEEvent has a unique Name
[ ] ZC_Service_*_Menu router is not duplicated
[ ] every consequence exists in XSD
[ ] specific hero effects are inside Companion
[ ] SelectedActor / SelectedFamilyMember / SelectedLord etc. are used instead
of Hero
[ ] relationships use RelationRef
[ ] pregnancy uses PregnancyTargetRef and PregnancyPartnerRef
[ ] LoadingFailedXML.txt has no errors
[ ] the event appears as Added in the log
```

---

## 24. Most Important zCapacity Rule

If an event is supposed to affect a specific character selected in the menu, always use:

```
<Companion Ref="Selected..." Type="Selected...">
```

Do not rely on default `Hero`.

zCapacity works best when the XML clearly defines:

who is the actor  
who is the target  
who receives the relationship effect  
who receives the consequence

This makes events predictable, easier to debug and safer to expand.