



národní  
úložiště  
šedé  
literatury

## **Instalace systému CDS Invenio**

Národní technická knihovna  
2011

Dostupný z <http://www.nusl.cz/ntk/nusl-111906>

Dílo je chráněno podle autorského zákona č. 121/2000 Sb.

Licence Creative Commons Uveďte autora-Zachovejte licenci 3.0 Česko

Tento dokument byl stažen z Národního úložiště šedé literatury (NUŠL).

Datum stažení: 06.08.2024

Další dokumenty můžete najít prostřednictvím vyhledávacího rozhraní [nusl.cz](http://nusl.cz) .

# Instalace systému CDS Invenio

Tento text popisuje postup instalace systému CDS Invenio v konfiguraci NUŠL. Tento software si spolupracující organizace nainstaluje na vlastním hardwaru a dále upraví dle vlastních požadavků při zachování odpovídajících struktur (sbírek, formulářů, ..) určených Národním úložištěm šedé literatury (NUŠL). Toto omezení se ovšem týká pouze šedé literatury, a tedy správa ostatních digitální objektů na lokálním úložišti je plně v režii dané organizace.

## Lokální instalace systému CDS Invenio

Každý zájemce o systém CDS Invenio si jej může volně stáhnout z <http://invenio-software.org/wiki/Installation/Download> a nainstalovat si ho podle návodu, což zahrnuje instalaci všech závislostí, nastavení, kompilaci a instalaci samotného systému, nastavení certifikátů, databáze atd. Pro spolupracující organizace s NUŠL je připraven instalační skript, který většinu těchto kroků vyřeší. Navíc provede i základní nastavení Invenia, takže je možné jej ihned používat. Instalaci je teoreticky možno provádět pod jakýmkoli operačním systémem linuxového typu, který jako správce softwarových balíčků používá Advanced Packaging Tool (apt, apt-get). Prakticky je instalace úspěšně vyzkoušena na systémech Debian 6 a Ubuntu 11. Jediný software potřebný ke spuštění instalačního skriptu je programovací jazyk Python, který však bývá v základní instalaci operačního systému.

Instalaci je nejlepší provádět na čerstvé instalaci OS. Je k tomu možné využít nějaký virtualizační nástroj jako například VirtualBox (<http://www.virtualbox.org/>), VMWare (<http://www.vmware.com/cz/>) či kterýkoli jiný.

## Instalační skript

Archiv obsahuje 2 soubory - invenio\_install a invenio\_install.cfg.

Soubor invenio\_install.cfg je konfigurační soubor instalace, editujte jej libovolným textovým editorem (např. nano, gedit, kate). Lze v něm nastavit adresář pro pomocné soubory instalace a hesla pro MySQL uživatele. **Důležité** je nastavit správně heslo pro MySQL uživatele root. Pokud již máte nainstalovaný databázový systém MySQL, zadejte heslo pro existujícího uživatele root. Pokud MySQL nainstalováno nemáte, zadejte libovolné heslo a při vlastní instalaci je nutné zadat stejné heslo, jinak instalace neproběhne úspěšně. Konfigurační soubor musí být při spuštění vlastního instalačního skriptu ve stejném adresáři jako instalační skript. Soubor invenio\_install je vlastním spustitelným skriptem instalace. Může být nutné změnit práva pro spuštění:

```
chmod u+x invenio_install
```

Instalace probíhá ve 3 krocích:

1. Instalace (instalace Invenio, všech závislostí, podpůrné webové služby, certifikátu aj.)
2. Nastavení parametrů (nastavení základní URL adresy, uživatelů a hesel MySQL aj.)
3. Nastavení systému (provede základní konfiguraci systému CDS Invenio - sbírky, indexy, formuláře aj. a aplikuje změny ve zdrojovém kódu z NUŠL repozitáře na Google Code)

Obě fáze (instalace a nastavení systému) jsou rozděleny do několika kroků, které lze v případě potřeby provádět i samostatně (například po neúspěšném provedení některého kroku).

Základní instalace se spouští s argumentem install:

```
./invenio_install install
```

Tato část instalace je interaktivní (většinou všude stačí dávat implicitní nastavení s výjimkou výše zmíněného nastavení root uživatele pro MySQL)

Po úspěšném provedení prvního kroku editujte soubor `/opt/invenio/etc/invenio-local.conf`, kde je nutné minimálně nastavit proměnné `CFG_DATABASE_USER`, `CFG_DATABASE_PASS`, `CFG_SITE_URL`, `CFG_SITE_SECURE_URL`. Dobré je také nastavit proměnnou `CFG_BIBSCHED_PROCESS_USER` na hodnotu jména uživatele, který smí spouštět Invenio procesy (implicitní uživatel je zde `www-data` čili každá akce by se musel spouštět jako `sudo -u www-data [akce]`). Také se vyskytly problémy s způsobené proměnnými `CFG_BIBRANK_SHOW_CITATION_LINKS`, `CFG_BIBRANK_SHOW_CITATION_STATS` a `CFG_BIBRANK_SHOW_CITATION_GRAPHS`, které je tak dobré nastavit na hodnotu 0. Poté editujte soubor `/opt/indexer/settings.py` kde nastavte proměnné `DATABASE_USER` a `DATABASE_PASSWORD`.

Nyní můžete přikročit ke 3. fázi - nastavení systému spouštěním instalačního skriptu s argumentem `set`:

```
./invenio_install set
```

Stručný návod je obsažen i v samotném instalačním skriptu. Vyvoláme jej spouštěním skript s parametrem `--help`:

```
./invenio-install --help.
```

## Poznámky k instalaci

- Ne všechny balíčky, které invenio teoreticky může využívat lze jednoduše nainstalovat. Proto se v logu mohou objevit zprávy o chybějících balíčcích. Všechny důležité by se však nainstaloval měly.
- Většina výpisů během instalace je odkloněna do logů. Některé balíčky však vyžadují lidskou interakci, a proto je u těchto balíčků celý výpis poslán na standardní výstup (do konzole). Nejedná se o chybový výstup.
- Konfigurační soubory Invenia jsou v zásadě 2 - ve složce `invenio/etc` - `invenio.conf` a `invenio-local.conf`. Myšlenka je taková, že `invenio.conf` obsahuje veškeré proměnné v defaultní konfiguraci a `invenio-local.conf` může obsahovat jen některé proměnné ve změněné konfiguraci. Systém primárně bere hodnoty ze souboru `invenio-local.conf` a ty hodnoty, které zde nenalezne, vezme ze souboru `invenio.conf`.
- Pokud změníte jakékoli hodnoty v konfiguračním souboru (`invenio-local.conf`) je nutné tyto změny do systému zavést aby se projevily příkazem: `/opt/invenio/bin/inveniocfg --update-all` Pokud dojde ke změně nastavení webové komunikace (např. změna IP adresy) je nutné vygenerovat nové konfigurační soubory pro Apache příkazem `/opt/invenio/bin/inveniocfg --create-apache-confs`. Při instalaci se toto provádět nemusí, protože to za vás udělá instalační skript.

# Provozování systému CDS Invenio

## Obecné informace

- Interní metadatový formát záznamů v Inveniu je MARC21 <http://www.loc.gov/marc/>. Konkrétní podobu tohoto formátu a použitá pole naleznete zde [http://nusi.ntkcz.cz/images/Sablony\\_v1.pdf](http://nusi.ntkcz.cz/images/Sablony_v1.pdf)
- Velká část HTML výstupu CDS Invenio je cacheována. Aby se projevily změny, je třeba

tuto cache obnovit spuštěním akce webcoll, popř. restartem webového serveru.

- Velká část konfigurace systému CDS Invenio a téměř všechny úlohy se provádějí z příkazové řádky
- Všechny úlohy probíhají přes plánovač BibSched ovládaný z příkazové řádky

## Základní moduly Invenia

### BibIndex

Modul BibIndex umožňuje nastavení a spuštění indexace záznamů. Indexy jsou libovolně nastavitelné, indexují se jednotlivá slova, dvojice slov a celé věty. V indexované pole nabízejí záznamy, které jsou k našemu dotazu blízké (jinak lze obdržet jen naprosto přesné zásahy). Indexy jsou také nutné pro pole použitá v pokročilém vyhledávání.

Modul BibIndex se nastavuje přes webové rozhraní.

Nejprve je potřeba vytvořit si logická pole. Ta se skládají z kódu pole, nepovinně z lokalizovaných názvů a seznamů MARC tagů.

[Hlavní stránka](#) > [Admin Area](#) > [Manage logical fields](#) > [Edit Logical Field](#)

### Edit Logical Field

#### Edit logical field 'abstract'

##### Menu

0. Show all
1. [Modify field code](#)
2. [Modify translations](#)
3. [Modify MARC tags](#)
4. [Delete field](#)
5. [Show field usage](#)

#### 1. Modify field code for logical field 'abstract' [?]

Field code  [Modify](#)

#### 3. Modify translations for logical field 'abstract' [?]

Language	Translation	
Česky (def)	<input type="text" value="abstrakt"/>	
English	<input type="text" value="abstract"/>	<a href="#">Modify</a>

#### 3. Modify MARC tags for the logical field 'abstract' [?]

##### Menu

[Add MARC tag](#)

[Delete unused MARC tags](#)

	Value	Comment	Actions
▼	520__9	Jazyk abstraktu	<a href="#">Details</a> / <a href="#">Modify</a> / <a href="#">Remove</a>
▲	520__a	Abstrakt	<a href="#">Details</a> / <a href="#">Modify</a> / <a href="#">Remove</a>

Takto definované pole již můžeme použít pro vyhledávání. Syntaxe je code:hodnota např. abstract:Tato práce\* nám najde všechny záznamy, u kterých jeden z tagů definovaných v poli abstract začíná slovy "Tato práce".

Index se skládá z názvu, seznamu logických polí, jazyka pro stemování (implicitně dostupná pouze angličtina, lze vypnout) a nepovinně z lokalizací a popisu. Index jako takový se pro vyhledávání nepoužívá, ale použijí-li se indexovaná pole, tak lze vyhledávat i blízké zásahy.

## Edit Index

### Edit index

#### Menu

0. Show all 1. Modify index name / descriptor 2. Modify translations 3. Modify index fields 4. Modify index stemming language 5. Delete index

#### 1. Modify index name. [?]

Index name

Index description

#### 2. Modify translations for index. [?]

Language	Translation
Česky (def)	<input type="text" value="keywords"/>
English	<input type="text"/>

#### 3. Modify index fields. [?]

#### Menu

[Add field to index](#)

[Manage fields](#)

Field
keyword <a href="#">Remove</a>
PSH <a href="#">Remove</a>

#### 4. Modify index stemming language. [?]

Index stemming language

## WebSearch

Modul WebSearch slouží ke konfiguraci sbírek a nastavení vyhledávání ve sbírkách. Novou sbírku lze vytvořit pomocí jejího unikátního jména. Po jejím vytvoření ji můžeme rovnou zařadit do stromu sbírek.

## WebSearch Admin

### Menu

0. [Show all](#) 1. [Create new collection](#) 2. [Attach collection to tree](#) 3. [Modify collection tree](#) 4. [Webcoll Status](#)  
5. [Collection Status](#) 6. [Check external collections](#) 7. [Guide](#)

#### Create new collection [?]

Default name

#### Operation successfully completed.

#### Attach collection to tree [?]

These collections are not in the tree, and should be added:

[nová sbírka](#) ,

Attach collection:

to parent collection:

with relationship:

Sbírku zařadíme pod jinou sbírku (hierarchicky nejvýš je vždy implicitní kořenová sbírka, jejíž

název se vždy musí shodovat s názvem repozitáře, který se definuje v konfiguračním souboru) se vztahem Regular nebo Virtual. Sbírky se vztahem regular se (implicitně) nacházejí na levé části obrazovky a sbírky typu virtual na pravé. Regular sbírky by měly obsahovat každý dokument a právě jednou, protože z nich se vypočítává celkový počet záznamů v repozitáři. Záznamy ve sbírkách v pravé části se tedy se sbírkami regular logicky kříží a může se křížit i pravá strana sama se sebou.

Modify collection tree: Digitální repozitář [?] Printer friendly version

#### Narrow by collection:

- ✗ Digitální repozitář
  - ✗ Vysokoškolské kvalifikační práce
    - ✗ Habilitační práce
    - ✗ Disertační práce
    - ✗ Rigorózní práce
    - ✗ Diplomové práce
    - ✗ Bakalářské práce
  - ✗ Zprávy
    - ✗ Zprávy z průzkumu
    - ✗ Grantové zprávy
    - ✗ Závěrečné zprávy z projektu
    - ✗ Průběžné zprávy z projektu
    - ✗ Statistické zprávy
    - ✗ Technické zprávy
    - ✗ Výzkumné zprávy
    - ✗ Výroční zprávy
  - ✗ Autorské práce
    - ✗ Monografie
    - ✗ Preprinty
    - ✗ Referáty
  - ✗ Firemní literatura
    - ✗ Katalogy výrobků
    - ✗ Návodů
  - ✗ Konferenční materiály
    - ✗ Postery
    - ✗ Programy

#### Focus on...:

- ✗ Digitální repozitář
  - ✗ Vysoké školy
    - ✗ Jihočeská univerzita v Českých Budějovicích
    - ✗ Akademie múzických umění v Praze
    - ✗ Akademie výtvarných umění
    - ✗ Vysoká škola ekonomická v Praze
    - ✗ Univerzita Karlova v Praze
  - ✗ Knihovny
    - ✗ Západočeské muzeum v Plzni
    - ✗ Národní knihovna
    - ✗ Národní technická knihovna
    - ✗ Národní lékařská knihovna
  - ✗ Vědecko-výzkumné organizace
    - ✗ Ústavy AV ČR
      - ✗ Archeologický ústav, Brno
      - ✗ Archeologický ústav, Praha
      - ✗ Astronomický ústav
      - ✗ Biologické centrum
      - ✗ Biotechnologický ústav
      - ✗ Biofyzikální ústav
      - ✗ Botanický ústav
      - ✗ Entomologický ústav
      - ✗ Etnologický ústav
      - ✗ Filosofický ústav
      - ✗ Fyzikální ústav

Na úvodní stránce se zobrazují první dvě úrovně sbírek (nepočítám kořenovou).

POZOR, pokud chceme i a pravé straně na úvodní stránce zobrazit 2 úrovně sbírek, tak sbírku na první úrovni zařadíme pod kořenovou sbírku se vztahem VIRTUAL, ale druhou sbírku zařadíme pod sbírku na první úrovni se vztahem REGULAR!

Po rozkliknutí nastavení kolekce se dostaneme k individuálnímu nastavení kolekce. Pozornost je třeba věnovat především možnosti č. 1 (Query). Sem zadáváme vyhledávací výraz, který do této kolekce zařadí záznamy (viz. kapitola BiblIndex). Vyhledávací výraz klasicky vypadá takto:

```
collection:rigorozni_prace
```

Kde "collection:" značí logické pole (definuje se v modulu BiblIndex) a "rigorozni\_prace" konkrétní hodnotu u záznamu v tomto poli. Máme-li tedy nastaveno, že logické pole "collection" odpovídá tagu 980\_\_a, tak všechny záznamy, které v poli 980\_\_a obsahují textový řetězec "rigorozni\_prace" (nedbá se na velikost písmen ani na národní znaky (háčky, čárky, přehlásky aj. - dle nastavení databáze), tedy "rigorozni\_prace" == "Rigorozni\_Prace" == "Rigorózní\_práce" ) budou zařazeny právě do této sbírky. Alternativně lze místo názvu pole použít přímo MARC tag:

```
980__a:rigorozni_prace
```

což samozřejmě funguje i při klasickém vyhledávání.

## 1. Modify collection query for collection 'Rigorózní práce' [?]

Query  [Modify](#)

Možnosti vyhledávání (vyhledávací pole), řazení a výstupní formáty se v Inveniu s nezměněným zdrojovým kódem musely nastavovat pro každou sbírku zvlášť. Pokud nastaveny nebyly pak byly použity implicitní hodnoty, které předpokládaly používání DEMO nastavení Invenia. Pokud jsme se nacházeli na výsledcích hledání (tedy v žádné sbírce), pak byly na pevně nastavena všechna definovaná pole (přesto, že ne všechna musela být v nějakém indexu, což je vyžadováno u rozšířeného hledání).

V NUŠL je toto upraveno tak, aby bylo možné tyto volby nastavit centrálně. Podřazené sbírky přebírají tyto možnosti od sbírek jim nadřazených (pokud to není u nich nastaveno jinak). Ve výsledcích hledání je použito nastavení kořenové sbírky. Jedinou výjimkou zde jsou výstupní formáty. U těch (pouze ve výsledcích hledání) lze jejich výčet regulovat přímo v nastavení formátů (je to jedna z možností nastavení výstupního formátu).

## 6. Modify search fields for collection 'Atlantis Institute of Fictive Science' [?]

Field actions (not related to this collection)

Go to the BibIndex interface to modify the available search fields

Collection specific actions

[Add search field to collection](#)

[Order search fields alphabetically](#)

	Search field	Actions
▼	anyfield	<a href="#">Remove search field</a>
▲▼	abstract	<a href="#">Remove search field</a>
▲▼	title	<a href="#">Remove search field</a>
▲▼	fulltext	<a href="#">Remove search field</a>
▲▼	Datum vzniku dokumentu	<a href="#">Remove search field</a>
▲▼	keyword	<a href="#">Remove search field</a>
▲▼	PSH	<a href="#">Remove search field</a>
▲	author	<a href="#">Remove search field</a>



## 8. Modify sort options for collection 'Atlantis Institute of Fictive Science' [?]

Field actions (not related to this collection)

Go to the BibIndex interface to modify the available sort options

Collection specific actions

[Add sort option to collection](#)

[Order sort options alphabetically](#)

	Sort option	Actions
▼	anyfield	<a href="#">Remove sort option</a>
▲▼	abstract	<a href="#">Remove sort option</a>
▲▼	title	<a href="#">Remove sort option</a>
▲▼	fulltext	<a href="#">Remove sort option</a>
▲▼	Datum vzniku dokumentu	<a href="#">Remove sort option</a>
▲▼	keyword	<a href="#">Remove sort option</a>
▲▼	PSH	<a href="#">Remove sort option</a>
▲	author	<a href="#">Remove sort option</a>

## 10. Modify output formats for collection 'Atlantis Institute of Fictive Science' [?]

Output format actions (not specific to the chosen collection)

Go to the BibFormat interface to modify

Collection specific actions

[Add existing output format to collection](#)

	Code	Output format	Actions
▼	hb	HTML brief	<a href="#">Remove</a>
▲	hm	MARC	<a href="#">Remove</a>

## WebSubmit

Modul WebSubmit nám umožňuje obecně operovat se záznamy. Jeho primární účel (podle názvu) je záznamy přidávat, ale teoreticky s nimi přes tento modul lze zcela libovolně manipulovat - editovat, mazat, vytvářet různá speciální rozhraní aj.

Se záznamy manipulujeme pomocí formulářů, které lze vytvořit přes webové rozhraní, ale pro složitější manipulaci je třeba napsat si vlastní funkce např. pro tvorbu komplexního interaktivního rozhraní či nestandardního zpracování dat.

Příklad - formulář pro tvorbu nových záznamů:

### 1. Příprava elementů

Nejprve se připravíme elementy, které budeme ve formulářích používat. Tyto elementy

odpovídají jednotlivým částem výsledného záznamu (název, autoři, abstrakt atd.). Pod nabídkou menu 'Available elements' se nám zobrazí všechny nadefinované elementy a můžeme přidat nové. Nový element definujeme jeho názvem, který je třeba zvolit rozvážně. Názvy nesmí obsahovat některé speciální znaky a mají omezenou délku. Název by měl nějakým způsobem jasně popisovat o jaký element se jedná, nař 'abstractCz' pro český abstrakt nebo např. MARC tagem '520\_\_a\_cze', pokud vám to více vyhovuje. Elementy jsou v podstatě libovolné HTML formulářové prvky, takže zvolíme to, co je vhodné pro danou položku (pokud chceme něčeho neomezený počet, např: neomezený počet autorů, použijeme prvek textarea a každý autor bude na novém řádku, autoři budou od sebe odděleni v pozdější fázi zpracování). Políčka 'modification text' a 'marc code' slouží pouze k editaci záznamů pomocí modulu WebSubmit, ale dle mého názoru je k editaci záznamů vhodnější používat dobře nastavený BibEdit.

## 2. Příprava akcí

Zde můžeme přidávat nové akce (akce jako přidat záznam, upravit záznam atd.). Jedná se však pouze o jakési logické rozdělení, nic jiného.

## 3. Definice typů dokumentů

Položka menu 'Available Document Types' nám ani tak nenabízí různé typy dokumentů jako spíše různé WebSubmit formuláře. U každého typu dokumentu si pak nastavíme jaké chceme provádět akce. Pro každý typ dokumentu a každou akci se definuje jiný formulář. Tento systém nemusí zcela odpovídat naší logice. Např. v NUŠLu chceme mít pro každý typ dokumentu (akce - Submit New Record), jiný formulář, ale samostatnou indexaci existujících záznamů (což je logicky akce) bychom museli definovat pro každý typ dokumentu zvlášť a po každé stejně, což je očividně velice nepohodlný způsob. Proto je indexace v NUŠLu vedena jako typ dokumentu.

## 4. Submission rozhraní

Submission rozhraní se skládá z jednoho a více elementů. Ke každému vloženému elementu náleží jeho popis, označení zda je povinný či volitelný, krátký popis (ve formuláři se nezobrazuje) a JavaScriptová kontrola (lze kontrolovat např. formát datumu, při nesplnění kontroly se při pokusu o přechod na další stránku zobrazí upozornění, kde se mj. používá krátký popis.)

Tlačítka pro přechod mezi jednotlivými stránkami / kroky bohužel musí obsahovat javascriptové procedury kde se stránky či kroky nastavují. Příklad:

```
<input type="button"
onClick="document.forms[0].step.value=1;document.forms[0].currentpage.value=2;document.forms[0].submit();" value="Pokračovat",
class="adminbutton"/>
```

## Field Details:

Details of the authors field as it appears at position 5 on Page 1 of the SBICOPYRIGHT Submission:

Page Number: 1  
Field Number: 5  
Field Name: authors  
Creation Date: 2011-03-01  
Last Modification Date: 2011-03-01

Field Label:

```
<br /><br /><span style="color: red;">*
</span>Autoři (jeden na řádek):<br /><i>Fyzická/é
osoba/y, která je primárně odpovědná za vytvoření
obsahu zdroje, tj. výzkumné zprávy, technické
zprávy autor (vedoucí) výzkumného záměru...
```

Field Level:

Mandatory ▼

Field Short Description:

dc:creator

JavaScript Check:

--NO CHECK-- ▼

Save Changes

Cancel

## 5. Submission funkce

Submission funkce zpracovávají data z formulářových prvků submission rozhraní, která jsou uložena v jednotlivých souborech, jejichž názvy odpovídají názvům elementů (to však může být změněno v user-defined elementech, pokud použijeme atribut name).

### Element Details:

#### Element Preview:

Vybrat soubor Soubor nevybrán

#### Enter Element Details:

Element Name: file  
Creation Date: 2011-03-01  
Last Modification Date: 2011-03-01  
Modification Text:  
Element Type: User Defined Input ▼  
Marc Code: 8564\_u  
Size (text elements): 60  
No. Rows (textarea elements):  
No. Columns (textarea elements):  
Maximum Length (text elements):  
Value (text/hidden elements):

```
<input type="file" name="file1" />
```

Pro každou submission proceduru je vytvořen dočasný adresář, kam se soubory s daty ukládají. Adresář je implicitně umístěn v `invenio/var/data/storage/running/[doctype]/[submission_number]/`.

## 6. JavaScriptové kontroly

Formát uživatelem zadaných dat může být kontrolován javascriptovými funkcemi, které pak mohou být při sestavování submission rozhraní přiřazeny k určitému elementu. Tyto funkce berou jeden argument (obsah formulářového prvku) a vracejí 1 (v pořádku) či 0 (špatný formát).

**Enter Action Details:**

Check Name: DateCheck

Creation Date: 2010-08-19

Last Modification Date: 2010-08-23

---

```
function DateCheck(txt){
  if (txt.match(/^\d{4}$/) || txt.match(/^\d{4}-\d{2}-\d{2}$/) || txt == ''){
    return 1;
  }
  else{
    alert("Správné formáty dat jsou: RRRR-MM-DD, nebo RRRR.");
    return 0;
  }
}
```

Check Description:

### Příklad:

Pro vytvoření šablony např. pro zasílání vysokoškolských kvalifikačních prací si vytvoříme nový typ dokumentu (např. THESES) a v něm si implementujeme akci Submit New Record. Pro každý typ dokumentu si můžeme navíc definovat různé kategorie, které však již mají společné formuláře. Každá akce v každém typu dokumentu má pak nastavitelné rozhraní a funkce. V rozhraní uživatel zadává data a funkce tato data zpracovávají. Submission procedura se skládá z jednoho a více kroků. V každém kroku by mělo nejdříve být vytvořeno uživatelské rozhraní (definované pomocí elementů nebo generované funkcí), po jehož odeslání se spustí definované funkce pro daný krok a přejde se k následujícímu kroku opět sestavením rozhraní.

Submission Page Details:		Function Name	Step	Score	View Parameters	Delete
Document Type ID:	THESES	Create_Indexing_Interface	1	10	<a href="#">view parameters</a>	<a href="#">delete</a>
Submission ID:	SBI	Create_Recid	2	10	<a href="#">view parameters</a>	<a href="#">delete</a>
Number of Pages:	1	Report_Number_Generation	2	20	<a href="#">view parameters</a>	<a href="#">delete</a>
Creation Date:	2011-03-08	Move_Files_to_Storage	2	30	<a href="#">view parameters</a>	<a href="#">delete</a>
Last Modification Date:	2011-07-29	Make_Record_Python	2	40	<a href="#">view parameters</a>	<a href="#">delete</a>
		Insert_Record	2	50	<a href="#">view parameters</a>	<a href="#">delete</a>
		Print_Success	2	60	<a href="#">view parameters</a>	<a href="#">delete</a>
		Mail_Submitter	2	70	<a href="#">view parameters</a>	<a href="#">delete</a>
		Move_to_Done	2	80	<a href="#">view parameters</a>	<a href="#">delete</a>

Page	View Page	Delete
Page 1	<a href="#">view page</a>	<a href="#">delete page</a>

Add a Function
Finished

Add a Page
Finished

Na obrázcích můžeme vidět modelový submission proces o 2 krocích. V prvním kroku je zobrazena stránka Page 1. Po vyplnění formulářů a odeslání se všechna data z formulářů uloží do souborů, jejichž název je shodný s názvy elementů (pokud to není přebito například definicí user-defined elementů a nastavením atributa 'name') do dočasné složky `invenio/var/data/storage/running/[doctype]/[submission_number]/` a následně proběhnou všechny funkce kroku 1 - v tomto případě pouze `Create_Indexing_Interface`, což je funkce, která vrací nové uživatelské rozhraní (indexace PSH).

Po dalším odeslání se opět veškerá data z formulářů uloží do souborů v dočasné složce a provedou se funkce z kroku 2:

1. Create\_Recid - v databázi se založí nový záznam a je získáno číslo tohoto nového záznamu
2. Report\_Number\_Generation - vytvoří se reference, což se projeví i jako název souboru se záznamem
3. Move\_Files\_To\_Storage - uploadované soubory se přesunou do úložiště (implicitně /opt/invenio/var/data/)
4. Make\_Record\_Python - z dat zadaných ve formuláři je vytvořen marcxml soubor. Make\_Record\_Python je přidaná funkce nahrazující původní Make\_Record. Make\_Record\_Python umožňuje ke konverzi dat používat Python se vším všudy namísto původního systému (Plain text-oriented mode modulu BibConvert)
5. Insert\_Record - vytvořený marcxml záznam zařadí k uploadu
6. Print\_Success - Zobrazí informaci pro uživatele o úspěšném vložení
7. Mail\_Submitter - pošle vkladateli e-mail s potvrzením o vložení
8. Move\_To\_Done - dočasnou složku se všemi daty zkomprimuje a přesune do složky s dokončenými.

## BibConvert

Pomocí modulu BibConvert mohou být záznamy libovolně konvertovány mezi různými formáty. BibConvert lze samostatně používat z příkazové řádky a samozřejmě je využíván i jinými moduly (např. při harvestování a následné konverzi záznamů).

BibConvert z příkazové řádky lze použít například takto:

```
bibconvert -c template.xsl < record.xml
```

kde template.xsl je cesta k transformačnímu souboru (pokud není zadáno absolutní umístění, předpokládá se adresář invenio/etc/bibconvert/config) a record.xml je záznam v souboru (přesměrován do standardního vstupu). Výsledek je pak poslán na standardní výstup.

V klasické instalaci je možné používat dva typy transformačních skriptů - XSLT 1.0 a 'Plain text-oriented mode' (popsaný v dokumentaci k Inveniu). V NUŠL instalaci je možné použít třetí typ - čistě python skript (musí implementovat funkci 'convert', která vrací výsledek).

Protože se v Inveniu často pracuje s MARCXML formátem, tak byla vytvořena sada pythoních tříd pro práci s MARCXML, které jsou součástí bibconvert\_python engine. Tyto třídy je možné naimportovat do libovolného Python skriptu a usnadnit si tak práci s tímto formátem. Byly vytvořeny třídy: Record\_set, Record, Field, Controlfield a Subfield, import pak vypadá například následovně:

```
from invenio.bibconvert_python_engine import Record_set, Record
```

Třídy jsou zdokumentovány standardním způsobem, takže dokumentaci je možné si vyvolat shellovským příkazem:

```
pydoc invenio.bibconvert_python_engine
```

## BibFormat

Modul bibformat nám umožňuje zobrazení v libovolném počtu různých formátů (zobrazení se

nemusi za všech okolností týkat záznamu jako takového). Tento modul se dá rozložit na tři části - výstupní formáty (output\_formats), šablony (format\_templates) a formátovací elementy (bibformat\_elements)

## Výstupní formáty

Jedná se o definice jednotlivých formátů. Nacházejí se na souborovém systému v adresáři:

```
invenio/etc/bibformat/output_formats
```

ale spravují se z webového rozhraní. Tyto definice představují výstupní formát jako celek. Každý formát má jednu základní šablonu. Volitelně lze používat šablon více pro různé účely.

## Output Format HTML detailed Rules

menu

0. [Close Output Format](#) 1. Rules 2. [Modify Output Format Attributes](#) 3. [Check Dependencies](#)

Define here the rules the specifies which template to use for a given record.

1	Use template	Poetry HTML detailed	if field	980__a	is equal to	POETRY	[?]	Remove Rule 1
2	Use template	Picture HTML brief	if field	980__a	is equal to	PICTURE	[?]	Remove Rule 2
By default use		Default HTML detailed						

Add New Rule Save Changes

Na obrázku můžeme vidět definici formátu 'HTML detailed'. Implicitní šablona je použita 'Default HTML detailed'. Pokud pole 980\_\_a nabývá hodnoty 'POETRY', pak je použita šablona 'Poetry HTML detailed', pokud pole 980\_\_a nabývá hodnoty 'PICTURE', tak je použita šablona 'Picture HTML detailed'.

## Šablony

Šablony tvoří jeden konkrétní výstup, jejich konfigurace se provádí přes webové rozhraní a nacházejí se v adresáři

```
invenio/etc/bibformat/format_templates
```

Šablony mohou být složeny z formátovacích elementů nebo alternativně lze použít jazyk XSLT. Tvorba šablon přes webové rozhraní je interaktivní, je zde výběr ze všech dostupných elementů a můžeme průběžně sledovat výsledek na konkrétním záznamu.

## Format Template Default HTML detailed

Menu  
0. [Close Editor](#) 1. [Template Editor](#) 2. [Modify Template Attributes](#) 3. [Check Dependencies](#)

Format template code [Hide Documentation](#) Elements Documentation

```
<script type="text/javascript" src="/js/jquery.min.js"></script>

<BFE_TITLE_MULTILINGUAL prefix="<strong>
<lang>
<en>Title:</en>
<cs>Název:</cs>
</lang> </strong>" suffix="<br />" default="" escape="0" />

<BFE_AUTHORS_NEW prefix="<strong>
<lang>
<en>Authors:</en>
<cs>Autoři:</cs>
</lang></strong>" suffix="<br />" default="" escape="0" active="yes" withRoles="yes"
contributors="yes"/>

<BFE_DATE
prefix="<strong>
<lang>
<en>Date:</en>
<cs>Datum:</cs>
</lang></strong>" suffix="<br />"/>
```

[Save Changes](#)

Preview

Content-type (MIME):  Language:  Search Pattern:  [Reload Preview](#)

**ATLANTIS INSTITUT FIKTIVNÍCH VĚD** admin :: [odhlásit](#)

Hledej Přidej Personalizace ▼ Nápověda Administration ▼

[Hlavní stránka](#)

**Název:** [Original] [Translation]  
Databáze koncertních sálů v Praze  
**Autoři:** [Koutná, Kateřina](#) ; [Štílec, Jiří](#) (advisor) ; [Bělor, Roman](#) (referee) ; [Havlík, Jaromír](#) (referee)  
**Jazyk:** cze  
**Nakladatel:** Akademie múzických umění v Praze. Hudební fakulta AMU. Knihovna  
**Abstrakt:** [cze] [eng]  
Text této práce je dokumentací k praktickému projektu databáze Koncertní sály, jenž je uveřejněn na internetu na adrese [www.prehled.net/koncertni-saly](http://www.prehled.net/koncertni-saly). Cílem této bakalářské práce bylo vytvořit nástroj napomáhající hudebním pořadatelům k vyhledání vhodného sálu pro jejich projekt. Databáze obsahuje technické detaily prostorů pro hudbu v Praze a některých koncertních síní v regionech a umožňuje vyhledávání podle nejrůznějších parametrů. Součástí textu je kromě popisu tvorby této internetové aplikace také část týkající se porovnání parametrů významných sálů v ČR a dále pasáž, týkající se zmapování netradičních prostorů pro provozování hudby v Praze.  
**Typ dokumentu:** Bakalářské práce

## Formátovací elementy

Formátovací elementy představují jednotlivé části záznamu (autoři, název, abstrakt, ...). Každý element je jeden soubor na souborovém systému v adresáři

```
invenio/lib/python/invenio/bibformat_elements
```

a je pojmenován jako `bfe_[název elementu].py`

Každý element musí implementovat funkci 'format', která bere 1 povinný argument - instanci třídy `BibFormatObject` (`invenio.bibformat_engine.BibFormatObject`) + `**kwargs` (těmi můžeme element parametrizovat) a vrací string s výsledkem.

Všechny formátovací elementy mají 4 implicitní parametry (prefix, suffix, separator, escape), které už se nikde nedeklarují.

Po vytvoření nového elementu na souborovém systému se tento ihned objeví v nabídce modulu `BibFormat`, není ho třeba již nikde deklarovat.

## BibKnowledge

Tento modul umožňuje vytvořit jednoduché mapování (klíč : hodnota), které pak můžeme přes API využít dále (například v BibFormat)

## WebAccess

Pomocí modulu WebAccess můžeme spravovat uživatelské účty a oprávnění. Správa oprávnění v Inveniu je založena na rolích (role-based). Základem tohoto systému je role, ke které se váže množina oprávnění.

Například založíme si roli 'VIPusers', která bude moci přistupovat do kolekcí s omezeným přístupem a stahovat soubory s omezením (budou jí přiřazeny akce viewrestrcoll a viewrestrdocs - akce se vybírají z rolety).

Když máme připravenou roli, musíme ji nějak přiřadit k uživateli. To lze buď přímo (uživatel XXX bude mít roli YYY) nebo nepřímo (firewall-like).

Firewall-like definice umožňují například přiřadit roli všem uživatelům nacházejícím se v určité síti (podle IP adresy a masky podsítě). Kompletní reference se nachází dokumentaci modulu WebAccess v Inveniu.

## Omezení přístupu ke sbírkám

Přístup ke sbírce může být omezen. Toto omezení se však neprovádí v modulu WebSearch ale právě v modulu WebAccess. Stačí u kterékoli role nastavit akci viewrestrcoll s příslušným parametrem (název sbírky) a přístup k ní mají pak jen uživatelé s příslušnou rolí.

## Omezení přístupu k plným textům

Omezení digitálních dokumentů je přímé. V databázi v tabulce bibdoc se jedná o sloupec 'status', v manažeru digitálních dokumentů to najdeme pod názvem 'Access' a v MARCu je to pole FFT\_\_r (při uploadu). K souborům s

## Ovládání repozitáře z příkazové řádky

Všechny spustitelné soubory jsou uloženy v invenio/bin. Abychom nemuseli pokaždé vypisovat cestu ke spustitelným souborům, je dobré si do proměnné v prostředí PATH přidat cestu k bin adresáři: export PATH=\$PATH:/opt/invenio/bin . Místo napčř. " /opt/invenio/bin/webcoll" spouštíme programy pouze "webcoll"

## Bibsched

Bibsched je plánovač akcí pro Invenio (podobný CRONu). Všechny procesy (upload záznamů, harvestování, indexace a mnoho dalších) probíhá přes něj. Zadáním příkazu bibsched (popř. bibsched monitor) se nám spustí interaktivní rozhraní v příkazové řádce. Plánovač má dva módy - manual a automatic. V automatickém se akce spouštějí ihned, pokud je fronta volná. Pokud už nějaká akce probíhá, tak se nově příchozí akce zařadí za ní. Automaticky se též spouštějí periodicky naplánované akce (např. každou hodinu, každý den, ...). V manuálním módu se mohou akce spouštět, pozdržovat a zabíjet ručně.



```

invenio1 : invenio
Soubor  Úpravy  Pohled  Záložky  Nastavení  nápověda
-----
ID      PROC [PRIO]  USER      RUNTIME      SLEEP  STATUS  PROGRESS
-----
45      webcoll      admin     2011-11-09 10:15:45  1h     WAITING
631     bibindex     admin     2011-11-09 10:15:45  1h     WAITING
504     oaiharvest   admin     2011-11-09 15:34:02  1d     WAITING
46      oairepositoryupdateradmin  2011-11-09 16:11:44  1d     WAITING
632     bibrank      admin     2011-11-09 16:11:58  1d     WAITING
477     inveniogc    admin     2011-11-15 15:18:14  7d     WAITING
-----
Automatic Mode | A Manual | [1,2,3 Display] | P Purge | [L Log] | [O opts] | [O quit] | 2011-11-09 09:29:27

```

## Webcoll

Webcoll slouží k obnově webcache. Pokaždé když dojde k nějaké změně kolekcí (nová struktura, přidané záznamy), musí být spuštěn webcoll, aby se změny projevyly i ve webovém rozhraní. Je dobré mít spuštění webcollu naplánované periodicky pomocí parametru s: webcoll -s1h jej spustí každou hodinu; webcoll -s30m každou půlhodinu atd.

## Bibindex

Tato akce spouští indexování záznamů. Aby se nějaký záznam mohl objevit v repozitáři, musí být nejprve zaindexován (tedy ještě před spuštěním webcoll). Tuto akci je opět silně doporučeno spouštět periodicky.

## Bibupload

Upload záznamů v marcxml formátu. Upload může probíhat v 5 módech:

- i (insert) vloží nový záznam (ten nesmí obsahovat pole 001)
- r (replace) přepíše stávající záznam (musí obsahovat pole 001)
- ir (insert or replace) pokud je záznam nalezen tak je přepsán, pokud ne, tak vložen
- a (append) musí obsahovat pole 001, pole jsou ke stávajícímu záznamu připojena
- c (correct) musí obsahovat pole 001, pole ve stávajícím záznamu jsou nahrazena poli v novém záznamu

Příklad:

```
bibuplad -c opravny_zaznam.xml
```

## OAIharvest

Spuštění harvestování. Parametry jdou nakonfigurovat ve webovém rozhraní, spouští se pak pouze s parametrem -r [název repozitáře ve webovém rozhraní] (a samozřejmě s možnými plánovacími parametry jakoé -s atd.).

## OIArepositoryupdater

Udatuje příslušnosti záznamů do OAI setů. Konfigurace OAI setů se provádí ve webovém rozhraní a tato akce by opět měla běžet periodiky (pokud vystavujete nějaký set).

## Lokalizace

Repozitář je lokalizován pomocí nástroje gettext. Ten využívá textové soubory s příponou .po (pro každý jazyk zvlášť, najdeme je v instalačním adresáři invenio-1.0-rc0/po/), kde jsou dvojice - klíč : překlad. Klíč je zpravidla výraz v angličtině.

Příklad:

```
msgid "Choose how you want to restrict access to this file."
msgstr ""

msgid "Add new file"
msgstr "více"

msgid ""
"When you revise a file, the additional formats that you might have "
"previously uploaded are removed, since they no longer up-to-date with "
"the "
"new file."
msgstr ""
```

msgid je tedy klíč a msgstr je překald.

V kódu vypadá překlad následovně:

```
_("String to be translated").
```

kde '\_' je funkce, která vrací příslušně lokalizovanou hlášku. Pokud není lokalizace hláška nalezena, je vrácen původní řetězec.

Pokud není funkce '\_' zavedena, musíme ji závest:

```
from invenio.messages import gettext_set_language
_ = gettext_set_language(lang)
```

kde 'lang' je aktuálně nastavený jazyk (většinou lze získat z nějakého objektu).

Pokud provádíme změny lokalizace ve zdrojovém kódu, musíme soubor .po aktualizovat - ideálně automaticky:

```
find /opt/invenio/lib/python/invenio/ -name \*.py -exec xgettext -o
```

```
cs.po -j --from-code=UTF-8 {} \;
```

(cesty k souborům a adresářům musí samozřejmě odpovídat realitě)  
Lokalizační soubor se ovšem nepoužívá v textové podobě a je jej potřeba zkompilevat (z  
výkonnostních důvodů)

```
msgfmt -o /opt/invenio/share/locale/cs/LC_MESSAGES/invenio.mo cs.po
```