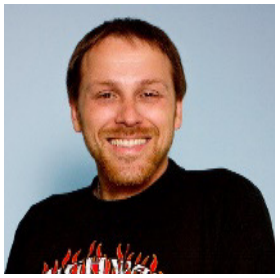
Advanced PDF/HTML Templates

aka Advanced Printing

NETSUITE | SuiteTraining

Presenter



Daniel Matyas

Platform Product Management, Team Lead

Cautionary Note: Forward Looking Statements

This presentation may contain “forward-looking” statements that involve risks, uncertainties and assumptions. These statements are based on information available to NetSuite management at the time of this presentation.

Actual results could differ materially from our current expectations as a result of many factors, including but not limited to risks associated with quarterly fluctuations in our business and results of operations; current macro-economic conditions and the possibility they could deteriorate; changes in business plans of our SuiteCloud partners and other strategic partners; and the effects of competition. These and other risks and uncertainties associated with our business are described in our most recently filed quarterly report on Form 10-Q for the most recent period end.

We assume no obligation and do not intend to update any forward looking statements.

Customers who purchase our services should make sure the decisions are based on features that are currently available. Please be advised that any unreleased services or features from NetSuite that are not currently available may not be delivered on time, or at all.

Agenda

We will look at:

- Advanced printing goals
- Template Editor
- Printing via SuiteScript
- Use case
- Resources

Goals

- Customize every aspect of printing
  - Basic Printing offers limited customization capability
- Print arbitrary record
- Printing framework for developers
- Replacement for Basic Printing
- Target audience
  - Business user with appropriate permission (no special knowledge)
  - Administrator (HTML and scripting knowledge)
  - SuiteScript developer

Template editor

NETSUITE | SuiteTraining

Demo content

- Feature activation
- List of templates
- Template editor
  - Visual mode
  - Source code mode
- Transaction form setup

Demonstration

Template editor

NETSUITE | SuiteTraining

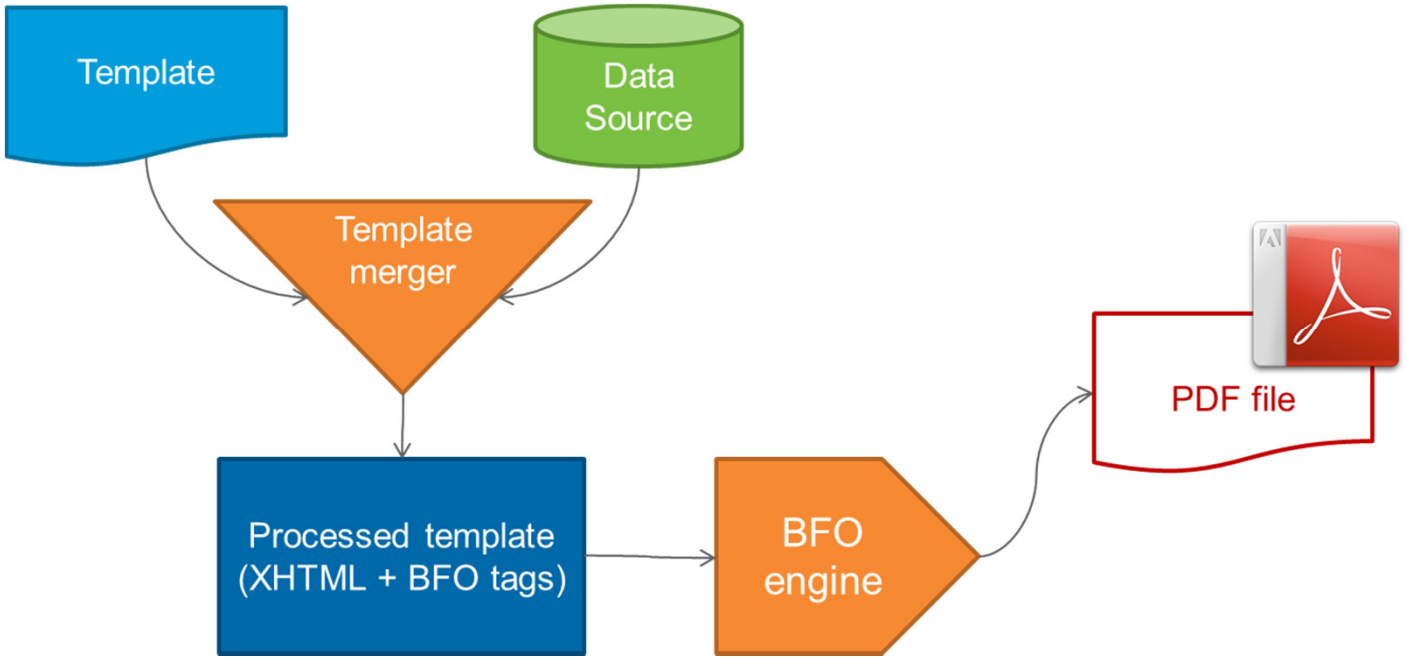
Printing via SuiteScript

NETSUITE | SuiteTraining

Advantages

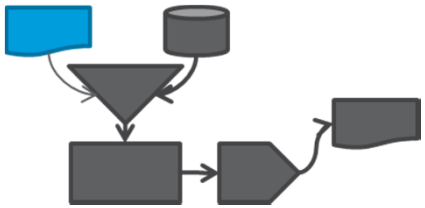
- Record types supported
  - Transactions
  - Entity records
  - Custom records
  - (Any record accessible via SuiteScript)
- Search supported
- Combination of records and searches
- Free format template

Architecture overview



Template I

- Syntax
  - XHTML, CSS2
  - FreeMarker elements
  - BFO tags
  - No SuiteScript
- Source
  - Script variable
  - NS File Cabinet
  - (Standard and customized template)

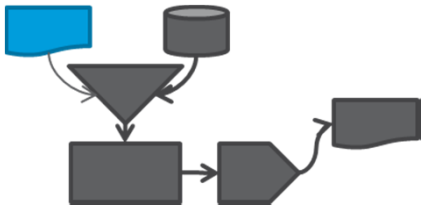


```
<?xml version="1.0"?>
<!DOCTYPE pdf PUBLIC "-//big.faceless.org//report" "report-1.1.dtd">
<pdf>
  <body font-size="18">
  ${record@title} ${record.tranid} <br/>
  ${record.total@label?upper_case} ${record.total}
  </body>
</pdf>
```

Template II

- FreeMarker

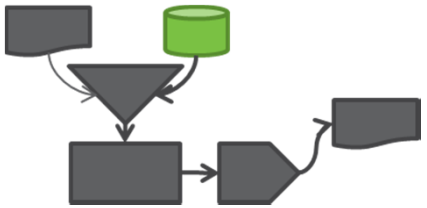
Record field	${record.address}
Record field label	${record.address@label}
Related record field	${record.entity.email}
Sublist field	${record.item[0].description}



- BFO
  - <pdf></pdf>
  - <pagenumber/>, <totalpages/>
  - … see user guide

Datasource

- Record
- Search
- Saved Search
- Combination of records and searches
- Synthetic record



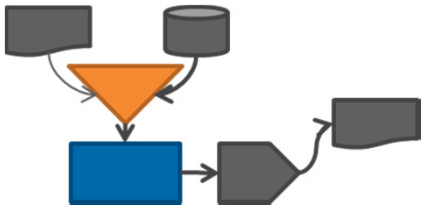
Example (SuiteScript):

```
var search = nlapiLoadSearch('transaction', 'customsearch165');
var results = search.runSearch().getResults(0,1000);

var employee = nlapiLoadRecord('employee','3');
```

Renderer

Example:



Initialize renderer

```
var renderer = nlapiCreateTemplateRenderer();

var fileCabFile = nlapiLoadFile('2208');
var template = fileCabFile.getValue();

renderer.setTemplate(template);
renderer.addSearchResults('graphdata', results);

var xml = renderer.renderToString();
```

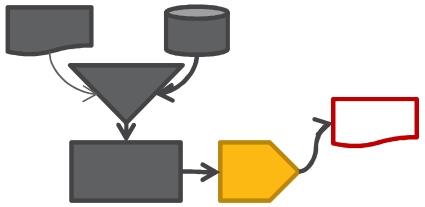
Load file from FileCabinet

Set template and data source

Process template

PDF creation

Example:



Generate PDF

```
var file = nlapiXMLToPDF(xml);

response.setContentType('PDF', 'sample.pdf', 'inline');
response.write(file.getValue());
```

Set PDF into response

Send response

Use case

NETSUITE | SuiteTraining

Scenario

Print employee record using custom template

- Employee basic data
- Expenses visualized as graph

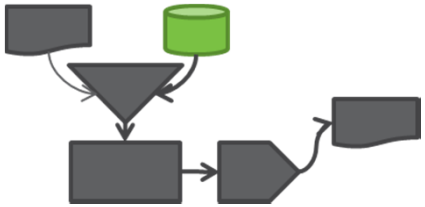
Datasource

Employee

- Employee record
- Standard NS record

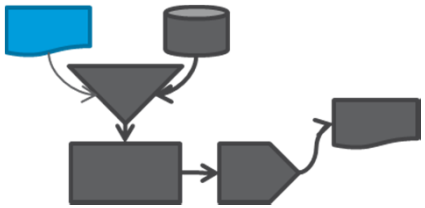
Employee expense list

- Expense Record saved search
- Custom built



Template

Employee



- Field label: `${employee.<field_name>@label}`
- Field data: `${employee.<field_name>}`
- Field names available in Help : Record Browser

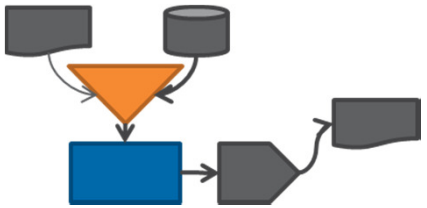
Employee expense list

- FreeMarker used to cycle through saved search result rows

```
<#list graphdata as row>
    ${row.<field_name>}
<#list>
```

Suitescript

Employee



Load record

```
var employee = nlapiLoadRecord('employee','3');
renderer.addRecord('employee',employee);
```

Add record as data source

Template alias

Employee expense list

Load search

```
var search = nlapiLoadSearch('transaction', 'customsearch165');
var results = search.runSearch().getResults(0,1000);
renderer.addSearchResults('graphdata', results);
```

Add search as data source

Template alias

Execute search

Demonstration

Use case

NETSUITE | SuiteTraining

Resources

NETSUITE | SuiteTraining

Resources

- Netsuite Help
- BFO userguide
  - http://bfo.com/products/report/docs/userguide.pdf
- FreeMarker manual
  - http://freemarker.org/docs/

Q&A

NETSUITE | SuiteTraining

Thank you!

NETSUITE | SuiteTraining