Custom Code Management
Custom Code Analyzer
Custom Code Analyzer

Analyze custom code for clones, modifications and links to SAP standard code

Scope
- Compare custom code against standard SAP source code and obtain the similarity degree
- Determine links between custom code and standard SAP Interfaces
- Determine modifications within a system

Benefits
- Find redundant and obsolete custom code in your system
- Determine the similarity of your custom code compared to SAP standard code, thus enabling to verify if the SAP code fulfills customer requirements and if a retrofit into SAP Standard is possible.
- Compare similarity between custom code across system borders
- Identify links between SAP Interfaces and customer objects, thus enabling identification of customer objects that might need to be adjusted during an upgrade or support package installation
- Determine the modifications within an SAP system
- Display of results in a simple ALV list, thus providing more options to analyze and customize the results
The “City Model” approach
The right Path for Continuous Improvement

Governance Model
Guidance for Decision-making and Execution

Quality
- Appropriate
- To be improved
- Inappropriate

Technical Severity

SAP Standard
Enhancement w/ interfaces
Enhancement w/o interfaces
Modification assisted
Modification not assisted
Custom with SAP reference
Custom (independent)

Quantity

Business Criticality
1: Non Mission Critical
5: Mission Critical

Custom Code Lifecycle Management
Custom Code Analyzer:
Use case 1: Reduce Quantity

How many custom code objects exist along the software lifecycle?
Custom Code Analyzer:
Use case 2: Determine Technical Severity

What is the technical severity of my implemented custom code?
(Deviation from SAP standard)

- **SAP Standard Configuration**
  - Customizing (e.g. IMG table entry)
  - Personalization (e.g. default values for user)

- **Enhancement** (SAP interface technology)
  - With interfaces
  - Without interfaces

- **Modification** (Changes to SAP Repository objects)
  - Assisted
  - Without any tools (freestyle)

- **Custom Code**
  - With reference to SAP Objects
  - Standalone and Composite Application
Custom Code Analyzer

Analysis of Custom Code

- Persistency...
- Expert Analysis

Custom Code Analysis - Use Cases

Start Custom Code Analysis in predefined Variants for special Use Cases:

- SAP Clone Finder: Find clones of SAP objects in your namespace
- SAP Interface Analysis: Detect dynamic links between SAP and custom code
- SAP Modification Overview: Get an overview of your SAP modifications
SAP Clone Finder

Reduce waste of clones

- Find clones in your own system landscape, across systems
- Determine the similarity degree of your clones against SAP originals and also inherited clone versions
- Fast and impressive overview in an easy to use UI to manage clones
- Direct split screen editor feature to merge clone differences
- Enhanced attributes to support the clone retirement process
- Statistical information about LOCs, versions and related business context areas
How does the Clone Finder work?

*Clones are determined in 5 steps*

- The SAP Environment of the Cloned customer object is determined Ex. LO-SCI*
- Packages with high chances of detection are identified based on the determined SAP Environment
- The Packages are analyzed and the original SAP objects are tokenised
- Customer program is tokenised and good candidates for comparison are detected „fingerprint detection“
- ‘Similarity Rate‘ is defined based on the number of matching tokens.
  If there is a 100% token match, a detailed (1:1) comparison is performed.
Interface Analyzer:
Use case: Determine ‘Dynamic’ link

**Static calls** can be traced using ‘Where used list’

- Ex. Go To transaction SE37
- Enter the name of the function module and use the option

```plaintext
REPORT z_DYNAMIC_CALL1.
data: l_name type char100.
call function 'z_GET_NAME'
   importing
      IM_NAME = l_name.
```

*In contrast* **Dynamic calls** cannot be traced!

```plaintext
report z_dynamic_call1.
data: l_name type char100.
data: function type funcname.
SELECT single function FROM swd_funcs
   INTO function where function like 'z%'.
call function function IMPORTING im_name = l_name.
```

The same function module `z_get_name` stored in customizing table swd_func and called dynamically called cannot be traced!
SAP Interface and Dynamic usage Analyzer

- Comprehensive overview of integrated customer & enhancement techniques (User Exit, Custom Functions, BAdI Implementation, Class inheritance,…)
- Find dynamic connections between SAP Code and custom code (Customizing, table entries)
- Fast and impressive overview in an easy to use UI
- Statistical information about LOCs, versions and related business context areas
Modification Overview

- Find modifications in your own system landscape, across systems
- Detect real modifications
- Statistical information about LOCs, versions and related business context areas
- Direct split screen editor feature
- Fast and impressive overview in an easy to use UI
Restrict the search area

Determine the level of similarity analyses
Support of transport elements

Own buffering engine with result reusage to optimize runtime
### SAP Clonefinder (patent pending)

#### Custom Code Similarity Analysis (to SAP)

<table>
<thead>
<tr>
<th>Obj Type</th>
<th>Customer Object Name</th>
<th>Size</th>
<th>Areas</th>
<th>Versions</th>
<th>Creation Date</th>
<th>GS</th>
<th>Reference</th>
<th>Text</th>
<th>Object Type</th>
<th>Interface Object Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Source Code</td>
<td>LZCC_BERLINTOP</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>No SAP Reference added</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Source Code</td>
<td>LZCC_BERLINUXX</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>No SAP Reference added</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Source Code</td>
<td>SAPLZCC_BERLIN</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>No SAP Reference added</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>ZJUNRO050</td>
<td>71</td>
<td>8</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>Very Similar Source Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>EMMA</td>
<td>178</td>
<td>24</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>Very Similar Source Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>Z_STOCK_OVERVIEW</td>
<td>258</td>
<td>24</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>Very Similar Source Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function Module</td>
<td>ZCHECK_MY_ARTCLASS</td>
<td>50</td>
<td>2</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>Very Similar Source Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function Module</td>
<td>Z_MATERIAL_MANTAIN_DARK</td>
<td>235</td>
<td>6</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>Very Similar Source Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function Module</td>
<td>Z_JUNRO050</td>
<td>21</td>
<td>21</td>
<td>1</td>
<td>31.08.2010</td>
<td></td>
<td>Very Similar Source Code</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Weighted Similarity in %

- Find clones also if naming is different
- Split screen editor

---

© 2011 SAP AG. All rights reserved.
Screenshots

Full overview about dynamic programming models
Forward navigation to the IMG customizing
Custom Code Analysis – Key take away

- Use Clone Finder to find clones and delete unused and irrelevant clones
- Use Interface and Dynamic reference analyser to detect used SAP interfaces and trace dynamic code calls
- Use 'Modification Overview' to get an overview regarding the Modifications and assess the Technical Severity

"Transparency is the first step to reduce the “TCO Effect” of custom code"