Custom Code Management

CCLM – Custom Code Lifecycle Management
(SP12-SP14)

SAP Active Global Support
Version 5.0
October 2015
CCLM – Custom Code Lifecycle Management

⚠️ This document is only valid from Solution Manager 7.1 SP12 or higher. It includes now as well new capabilities delivered with SP14.

From SP10 the set-up of UPL is part of the Solution Manager Set-up and from SP11 onwards the set-up of UPL and CCLM is integrated in the Solution Manager Set-up.

For the set-up of CCLM this guide can be used as well from SP11 onwards.

For the features of CCLM described in this document SP12 is the basis and can only be used from SP12 onwards. Some of the features already show capabilities which will be delivered with SP14.

The documentation for the implementation of CCLM from SP11 onwards is in addition directly embedded in the Solution Manager set-up. Please ready the documentation in the respective UI carefully.
Introduction

This document will give you a short introduction into the Solution Manager 7.1 application SAP CCLM.

It is a reality that SAP standard functionality does not fulfill 100% of every customer needs. As a result, SAP customers tailor their SAP applications to fit specific needs, a process that is mainly done at implementation time, but that it also continues to emerge as the business needs change.

With time, those custom developments may become obsolete due to new SAP functionality implemented via new versions or support packages, also sometimes because the requirement was desirable but not really needed or because the requirement was not what users requested. All these reasons result in a good part of custom code to remain unused and in some cases, unknown to system administrators.

Custom Code Lifecycle Management (CCLM) was designed to manage ABAP developments along the entire lifecycle of custom code objects from the creation of objects, to their use in productive systems all the way to clearing of unused custom code objects.

CCLM makes visible all ABAP developments and provides the capability of seeing them in conjunction with the 3+1+1 dimensions of the city model: quantity, technical severity, business criticality, + quality, + usage.
Table Of Content

What is Custom Code Lifecycle Management (CCLM)? .................. 6
Benefit Cases................................................................. 6
Use Cases: ................................................................. 6

How CCLM works..................................................................... 7
  CCLM high level architecture............................................. 7
  CCLM system pre-conditions............................................. 8
Main CCLM use cases.......................................................... 8

CCLM Set Up ........................................................................ 9
Custom Code Management in Solman_Setup Solution Manager 7.1
  (SP11 and higher)........................................................... 9
Collecting Usage.................................................................. 13
SAP Online help documentation........................................ 14
Implementation of Notes & Corrections .................................. 14
Maintain the Authorizations.................................................. 15

Defining the CCLM Library ...................................................... 15
Library Definition................................................................... 15
  Library Object Classes..................................................... 16
  Attributes of the Objects Classes....................................... 16
  Objects & Assigned Attributes for Object Classes................. 17
  Implementing the Library Definition.................................... 17
Maintain the CCLM Landscape Setup........................................ 17
Maintain the SRCSYSTEM Mapping (optional)......................... 18

Set up the Collector Jobs.......................................................... 18
  Choose a Lead System...................................................... 18
  Deletion of Objects.......................................................... 19
  Activate Local Objects (optional)......................................... 19
  Set-up of managed systems................................................. 20

Add or delete customer-specific attributes............................. 20
Define customer specific values........................................... 22
Extract data for customer-specific attributes via customer-specific
  collectors................................................................. 22
Define customer specific actions on custom code objects........... 24
Define customer specific GOTO on custom code attribute classes.. 25
Customer specific CDP determination................................... 25
Maintain Organization Units for the contract/owner maintenance .... 26

CCLM - How to use .................................................................. 27
Main User Interface.............................................................. 27
  Overview................................................................. 28
  Library Definition........................................................ 33
  Objects................................................................. 33
    Object classes......................................................... 34
    Custom Code Object selection and details......................... 34
    Standard attributes.................................................. 35
    Maintainable attributes.............................................. 35
    Filter objects by..................................................... 35
Search objects with attributes ................................................................. 36
Object type PROG .................................................................................. 36
Owner Concept ...................................................................................... 36
Contract Concept .................................................................................. 37
Add reference ......................................................................................... 38
Maintain attributes ................................................................................ 40
Mass Maintenance of objects ................................................................. 40
Quality ....................................................................................................... 40
Decommissioning .................................................................................. 41
  Decommissioning Cockpit ................................................................. 42
  Create New Analysis .......................................................................... 43
  Analysis List ....................................................................................... 45
Dashboards ............................................................................................ 47
  Business Value ................................................................................... 48
  Key Performance Indicator ............................................................... 48
  Input Measures .................................................................................. 51
  Quick View ......................................................................................... 52
BW Reporting ......................................................................................... 53
Common tasks ....................................................................................... 55
  Call Ad-hoc reporting ....................................................................... 55
Related Links .......................................................................................... 58
  Custom Code Transparency ............................................................... 58
  CDMC ................................................................................................. 58
  Custom Code Quality ......................................................................... 59
  ABAP Code Inspector ....................................................................... 59
  ATC Results ....................................................................................... 59
  ATC Open Exemptions ..................................................................... 59

**Criticality (Solution Manager Set-up -> Custom Code Management -> Step 8)** ................................................................. 61

Select the data pools (step “Analysis”) .................................................. 61
Define weightage for each data pool (step “Apply Weightage”) ................. 62
Schedule SOLAR project (step “Project Analysis”) .................................. 63
Schedule criticality job (step “Schedule Job”) ........................................ 63

**Troubleshooting / Q&A** ................................................................... 64

**Appendix** ......................................................................................... 70

**Special customizing of CCLM** ........................................................... 70
What is Custom Code Lifecycle Management (CCLM)?

The management of custom code supplements tools already available in SAP Solution Manager such as the Custom Development Management Cockpit (CDMC). CCLM was developed especially for the purpose of accompanying your ABAP enhancements and new developments throughout their whole lifecycles. This cycle begins when you create an object (program, transaction, table, class, etc.), followed by its use in production systems and extends through the retirement of the object in case of non-use or a reorientation of the development.

This newly developed application is a part of the SAP Solution Manager 7.1. The heart of the application is a generic library model with which you can classify and manage custom code objects developed by your organization. Information about the use of these objects, their quality and version-history in the connected systems can also be collected if desired. You get an application that provides instant transparency for your custom code and records its use in a complex landscape without any additional manual effort.

The generic central library is used by SAP as the central data source for all information on customer objects. You benefit particularly from the possibilities of individual assignment of responsibilities and contracts, consolidation of developments within an organization, and total control over new developments. It is possible to assign any object or list of objects to a contract or other predefined attributes.

The objective in introducing this application is to ensure that you can achieve the best possible adaptation of custom code to SAP code and therefore receive the best possible support. You can also benefit from cost savings in possible upgrades to higher SAP releases.

Benefit Cases

- **Cost reduction** Reduce the effort for an upgrade and other projects by setting the focus on relevant custom code objects
- **Control and Simplification** Governance and cleansing activities through providing of ownership, contract and other information
- **Sustainability** Automatic refresh makes an update of master data, maintainable data will be untouched
- **Transparency and Awareness** Find out what objects are used in which landscape and with a certain version at ‘high’ quality

Use Cases:

- Detect used custom code in complex SAP landscapes
- Document ownership, contract and any other information for custom code objects
- Automatic refresh and maintenance mechanism of master data, maintained information will be untouched
- Transparency of custom code usage, quality and version in different systems
How CCLM works

The set-up of CCLM in the SAP Solution Manager is done in two main steps:

**Step 1:** Call transaction solman_setup and define the library and start the extractor and collector jobs. Data is extracted and saved to the library periodically. The scheduled programs ensure that information is collected and written or overwritten in accordance with the active library definition.

**Step 2:** Call transaction sm_workcenter or CCLM to maintain the data and to work with the objects in CCLM. Some of the attributes can be maintained manually. The information entered by you manually is not overwritten and is retained long-term. The attributes for your individually developed objects that are not collected automatically, such as a lifecycle status or distribution rule, and particularly references to contracts and responsible persons, must be maintained manually. For more details about maintaining customer objects, see the attributes section below.

Additional manual steps are possible in the analysis and reporting area. Depending on your installation of an SAP NetWeaver Business Warehouse (BW) with the respective content, Web templates are provided. You can also make your own reports. Reporting and retention of historical data for use in the BW simplify the decision regarding the deletion or consolidation of custom code.

**CCLM high level architecture**

CCLM is an application that collects periodically data from managed systems, thus providing up to date information about custom code. Multiple landscape and systems can be configured to get data from. The information is saved locally on Solution Manager internal tables, thus allowing the quick display of figures without having to log on to the individual systems or wait for the information to come from them.
CCLM system pre-conditions

Custom Code Lifecycle management is available with Solution Manager 7.1 ST SP3 support package SP5 is recommended. For the features described in this guide SP12 or higher is needed.

Managed systems should have software component ST-PI 2008_1_x SP4 is needed, *SP5* is recommended.

Please implement all relevant notes and corrections in Solution Manager and managed systems (see chapter "Implementation of Notes & Corrections" below).

Attention: The presentation of the "City Model" in the overview screen and the "ICI dashboards" are developed with SAPUI5 technology. These screens will only run with Internet Explorer (IE) 10 or higher, Firefox and NWBC 4.0. If the presentation of the city model and the ICI dashboards are not working (message: "no SVG support") when starting the workcenter with transaction SM_WORKCENTER, please try to use transaction CCLM instead. Internet Explorer 9 should work with transaction "CCLM".

If it does not work for IE9 or IE8, there is a note from Microsoft:
Note: Windows Internet Explorer 8 does not natively support SVG and thus requires a plug-in to render SVG.


For BW reporting capabilities please install ST-BCO 7xx.

For Solution Manager SP12 you need to install ST-BCO SP11 to implement three new queries (0SM_ATC_RS_DETAILS; 0SM_ATC_DAILY_DETAILS; 0SM_ATC_RUNDATE_LOOKUP).

Main CCLM use cases

- Overview of custom code
- Distribution of custom code in relation to business owner
- Distribution of custom code in relation to contracts
- Evaluation of custom code objects in the landscape
- Usage analysis of custom code
- Evaluation of custom code quality
- Decommissioning of custom code
CCLM Set Up

Custom Code Management in Solman_Setup Solution Manager 7.1 (SP11 and higher)

⚠️ Attention: Please be aware the set-up has to be executed if you want to newly implement CCLM. It has to be executed as well if you have already implemented CCLM in a SP lower than 12 in order to update the existing CCLM implementation.

With the shipment of Support Package 11, Custom Code Lifecycle Management will be part of the SOLMAN_SETUP with an own setup procedure. This workcenter with a guided step by step procedure will replace the set-up as it was designed in SP10 or lower SP.

For the set-up of CCLM the information provided in this guide can also be used from SP11 onwards.

The set-up for CCLM is part of Custom Code Management Set-Up with the main parts for CCLM in step7.

Detailed information about the set-up is embedded directly in the UI for the set-up.

⚠️ Attention: Please be aware before setting up CCLM in steps 7.1 and 7.2. The complete steps 1-6 have to be executed.

⚠️ Attention: The settings of step 7.1 are system independent! If some collectors should not run in some systems, then they should not be scheduled for these systems in step 7.2.

CCLM - How to setup 9
**Attention:** To get the quality data into CCLM an ATC master run must be scheduled on the managed system. This means the new and strategic ATC tool must be implemented in the respective system for all new CCLM installations from SP11 onwards. For existing CCLM installation Code-Inspector results can be considered if the collector configuration for the quality collector in the Solman Setup for Custom Code Management step 7.1 is set to "RFC to Managed System". It is recommended to implement ATC and use the "local" mode in the set-up.

<table>
<thead>
<tr>
<th>Collector Configuration</th>
<th>Active</th>
<th>Read Mode</th>
<th>Customer Solution Function Mod</th>
<th>Execute customer solution on Managed System (Via RFC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects Extractor</td>
<td>✓</td>
<td>local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Extractor</td>
<td>✓</td>
<td>local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Criticality</td>
<td>✓</td>
<td>RFC to Managed System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Similarity</td>
<td>✓</td>
<td>Customer Solution</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the collector configuration (step 7.1) "local" means that the data for CCLM will be collected not directly from the manged system, but out of the new BW cubes of Solution Manager BW. "RFC to managed system" means that it is collected directly from the managed system, as it was defined before SP11, without using BW of Solution Manager. The option "RFC to managed system" is only available for existing CCLM implementations (implementations before SP11), all new implementation can only use the option "local". For all implementations it is recommended to select "local".

The whole set-up for Custom Code management is divided into 8 steps as part of a guide procedure.

Step 8 defines the criticality calculation it will be described later in this document. Step 7 is the main step to define the library.

The steps 1-6 are defining the basic set-up for Custom Code Management. These steps are also needed for other solutions e.g. SEA.

Step 1 is preparing the SolMan system and is executing up all necessary BW activities.
Step 2 (optional) is setting up housekeeping activities for BW cubes for SQLM, ATC and UPL.
Step 3 is creating roles, template users and authorizations for Custom Code Management.
Step 4 is defining the managed systems, which shall be monitored for Custom Code Management.
Step 5 is defining the client, which shall be used for the different managed systems.
Step 6 is starting the extractors, which are needed to fill the BW cubes for Custom Code Management.
**Attention:** If you want to activate housekeeping settings in step 2 for a specific system and a specific info cube, please be aware that there is no "Deactivation" button. Any time you activate the housekeeping settings for a selected system and a specific info cube the set-up deactivates the housekeeping settings for all not selected systems for this specific info cube! To avoid deactivation always select all system for this specific info cube where you would like to activate the housekeeping settings, also those ones again, which show already green lights for this info cube.

**Hint:** Step 1 for UPL: Here the system is only checking if BW content for UPL is activated. The activation of UPL content itself is part of "Basic Configuration" step 5 "Configure Automatically" activity: "BW Content Activation (UPL)".

**Hint:** Step 5 client selection: The client is needed to select the right RFC with needed authorization to analyze and collect the data from the managed system. The data, which will be collected or analyzed, is client independent.

**Hint:** Step 6.2 is only scheduling the extractor jobs. To check if an extractor run was successful please open the extractor framework. To open the extractor framework UI just click on the link "Extractor Framework Admin UI" in step 6.2 on top of the table listing the systems. In the extractor UI you can filter for the system and the name *cust* to check the custom code extractors. Every extractor runs through different phases. You can see them if you select an existing extractor run in the table below the list of extractors. The extractor phases will be listed below the runs in a new table. At least three phases are needed "Extractor", "Dataloader" and "Finalizing". To check if an extractor run was successful all phases have to appear in the table and all phase must show a green light. These extractors need a Read-RFC connection. In addition in this step the extractor can only be activated. It is not possible to deactivate the extractors in this step. To deactivate the extractors please change to the extractor framework UI.

**Hint:** Step 6.3 is needed to schedule the analysis jobs to prepare the managed systems and the data for the extractors for UPL, References and Similarity. For UPL a job is scheduled in the managed system to prepare the UPL data for extraction to Solution Manager. This jobs also starts UPL if UPL was not active before. Without this job no data will be extracted. These analysis jobs need a TMW-RFC connection, as they need to read and write data in the managed system. The references job analysis the references of an object, this is needed to fill the 0SM_CCREF cube to determine for example the DerivedApplication Component attribute. The 0SM_CCREF cube is also needed to get information on object references to for example derive usage information to referenced objects. The similarity job is needed to fill the similarity attributes for CCLM and the lines of code in CCLM.
In the custom code set-up the following extractors will be scheduled in each managed system to extract the data for Custom Code Management:

<table>
<thead>
<tr>
<th>extractor name (new)</th>
<th>description</th>
<th>cube/table</th>
<th>extractor name (according to step 6.2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Code library Extractor</td>
<td>extracts CCLM data into the generic CCLM cube</td>
<td>0SM_CCLM</td>
<td>CCLM</td>
</tr>
<tr>
<td>Custom Code UPL Extractor (MultiCube Architecture)</td>
<td>extracts UPL data from managed system into BW cube</td>
<td>0SM_UPL, 0SM_UPL_W, 0SM_UPL_M, 0SM_UPL_Y</td>
<td>UPL (Cubes ..._W, ..._M, ..._Y will be updated by housekeeping jobs)</td>
</tr>
<tr>
<td>Custom Code Object Extractor</td>
<td>extracts custom code objects from managed system into BW cube</td>
<td>0SM_CCLG</td>
<td>CC Gen</td>
</tr>
<tr>
<td>SAP References in Custom Code</td>
<td>extracts custom code references from managed system into BW cube</td>
<td>0SM_CCREF</td>
<td>CC Ref</td>
</tr>
<tr>
<td>ATC monitoring for Custom Code (results)</td>
<td>extracts ATC results from managed system into BW cube</td>
<td>0SM_ATC</td>
<td>ATC</td>
</tr>
<tr>
<td>ATC monitoring for Custom Code (exemption)</td>
<td>extracts ATC exemptions from managed systems into BW cube</td>
<td>0SM_ATCEX</td>
<td>ATC</td>
</tr>
<tr>
<td>Custom Code City Extractor</td>
<td>extracts CCLM data into City Model cube</td>
<td>0SM_CCITY</td>
<td></td>
</tr>
<tr>
<td>Custom Code SQLM Extractor</td>
<td>extracts SQLM results from managed system into BW cube</td>
<td>0SM_SQLM</td>
<td>SQLM</td>
</tr>
<tr>
<td>Extractor for Custom code similarity</td>
<td>extracts similarity results (clones) from managed system into similarity table of CCLM</td>
<td>AGS_CC_SIMILARITY</td>
<td>CC Similarity</td>
</tr>
<tr>
<td>iCI CCM Extractor</td>
<td>extracts CCLM data into iCI cube for dashboard</td>
<td>0SM_ESRSG, 0SM_ICI*</td>
<td></td>
</tr>
</tbody>
</table>

In addition the cube 0CCMAW will be read to collect data from workload statistics.

All extractors are registered in table E2E_ACTIVE_WLI.

💡 **Hint:** If you need more information about sizing with regards to UPL data, please read SAP note 2159129.

⚠️ **Attention:** CCLM is now using the ICI dashboards. Therefore the Measurement Platform Set Up has to be performed as well. This is also a part of the Solution Manager set-up.

⚠️ **Attention:** CCLM is now using the SAP UI5 technology. Therefore the System Preparation step 5.5 "Configure Gateway" in the Solution Manager set-up has to be performed as well.

The set-up can be executed by the Solman-Admin User. It is also possible to restrict the set-up to the different configuration scenarios, e.g., Custom Code Management.
Collecting Usage

With SP12 the collection for usage changed. For CCLM only the date of last usage will be collected into table AGS_CC_LAST_USG. Before SP12 all usage was collected into CCLM table AGS_CC_USAGE. This table is now obsolete. All detailed usage information will now be recorded in BW cubes.

It is recommended to only use the last usage collector. All customers starting with SP12 can only start the last usage collector based on workloadstatistic and UPL. Please see screenshot below from SolMan set-up for Custom Code in step 7.1.

Customers who have already implemented CCLM before SP12 do have the possibility to additionally collect detailed information into table AGS_CC_USAGE with the respective daily (UPL), weekly (EWA) and monthly (EWA) collectors. It is not recommended to collect usage into AGS_CC_USAGE, as this table will no longer be supported by SAP. Please see screenshot below from SolMan set-up for Custom Code in step 7.2.

⚠️ Attention: It is currently not possible to collect usage data for all object types in CCLM. If not used objects will be analyzed please check carefully if usage data can be collected or not. For example for webdynpro (WDYx) objects and tables it is not possible to collect usage data.
For the following objects types usage can be collected directly or based on dependencies. This is as well only valid for objects in CCLM which are of SAPSeverity = 5. This is the case for the following objects:

- **PROG**: only for executable programms usage can be collected from UPL or in some cases from workloadstatistics.
- **PROG**: for includes usage can be collected from SP14 based on UPL derived from main object: If the respective main object is marked as used, all includes of this main object will be marked as well.
- **METH**: Usage of methods can be collected from UPL.
- **CLAS**: Usage of classes can be collected based on UPL derived from assigned methods: If any method of a class is marked as used, then the respective class will be marked as used as well.
- **FUNC**: Usage of function modules can be collected from UPL.
- **FUGR**: Usage of function groups can be collected based on UPL derived from assigned function moduls: If any function module of a function group is marked as used, then the respective function group will be marked as used as well.
- **DEV**: Usage of development classes can be collected based on UPL derived from assigned objects: If any object of a development is marked as used, then the respective development class will be marked as used as well.
- **TRAN**: Usage of transaction can be collected from workloadstatistics. (UPL can not record usage of transactions)

⚠️ **Attention**: In all SPs lower than SP14 it is not possible to collect usage information for includes, because during runtime the includes are part of the main program. These means includes will always be counted as unused objects, although they might be executed!

⚠️ **Attention**: For enhancements (SAPSeverity = 1) and modifications (SAPSeverity = 3), the usage collection is also not possible for some of the object types listed above. Especially derived usage calculation (e.g. for includes marked as enhancement or modification) is not possible. Please check carefully usage lists of objects with SAPSeverity = 1 or = 3).

**SAP Online help documentation**

It is recommended to spend some time in the documentation to get familiar with the key elements of CCLM work center.

The official online help can be found under [http://help.sap.com/saphelp_sm71_sp12/helpdata/en/3f/f184533b5ae547e10000000a41470/content.htm](http://help.sap.com/saphelp_sm71_sp12/helpdata/en/3f/f184533b5ae547e10000000a41470/content.htm)

**Implementation of Notes & Corrections**

Implementation of all relevant notes for Solution Manager can be done via collective note in Solution Manager set-up Basic Configuration step 4.

The central note for Custom Code Management in Solution Manager: 1969109
For the managed system please implement all relevant notes from the Service market place with search term CCLM under application component SV-SMG-CCM-CDM and SV-SMG-CCM.

This is recommended to avoid any issues and get also the latest available improvements and enhancement shipped via Support Packages.

Furthermore we recommend notes for Software component ST-PI on Managed systems or systems to be observed.

For the managed systems it is possible to check missing notes as part of the set-up step 6.1. (Solution Manger set-up - Custom Code Management - step 6.1). In this step the following will be checked: If the managed system is a Solution Manger then it checks missing implementation of notes in ST and ST-PI, if the managed system is not an Solution Manager it checks missing notes in ST-PI.

Or ensure you have installed the latest available support package for your Solution Manager 7.1

**Custom Code Lifecycle Management** is now fully integrated in the RTCCTOOL framework.

Please follow the details in [1906451](#)

---

**Maintain the Authorizations**

"Reports View" can be enabled. To access existing BW web templates you need additional authorizations:

<table>
<thead>
<tr>
<th>Authorization Object</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM_BPM_BI</td>
<td>SMWORK_BI</td>
<td>10 = Display (Display)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 = Change (Insert and change Webtemplate assignments)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30 = Delete</td>
</tr>
</tbody>
</table>

---

**Defining the CCLM Library**

⚠️ **Attention**: Some of the screen shots in the text below are still taken in a support pack lower than SP12, or might have meanwhile been changed slightly. These screen shots still show how CCLM is working although the UI might look different in some areas, which are not relevant in the specific case, which should be illustrated.

---

**Library Definition**

The core of CCLM software is a generic library definition that classifies custom-code objects and contains the set of information that can be collected. The data collectors then retrieve custom code attributes from the connected systems (also called managed systems) automatically. The main advantage of the data collectors is that they need to
be set up only once as a periodic job, which means that after the initial set up, the information continues to be automatically retrieved, thus providing always and up-to-date status without any additional effort.

These attributes and the relationship among them are located in the heart of the application, which is called “the library”. This generic library model allows the classification and management of data about all ABAP-based custom code objects that are by definition not SAP standard.

**Library Object Classes**

The object classes are the main entities in the library. Data attributes about these objects is collected and/or maintained. All custom code object types are considered either a Custom Code Object (for instance, programs, function groups and classes) or a Custom Code Duplicate.

Custom Code Items are only available until SP11. From SP12 onwards Custom Code Items are included in Custom Code Objects/Duplicates.

**Attributes of the Objects Classes**

Attributes are special fields or characteristics that we want to get information about objects. Examples of attributes are correction numbers, creation date and package. Here it is also defined if the object class could be maintained manually or not.
Objects & Assigned Attributes for Object Classes

In this tab we can see the relationship between objects and attributes. For instance, for Custom Code Objects we can see that attributes Deletion Flag, Development Class and Master Language can be collected or maintained.

The generic central library is used by SAP as the central data source for all information on customer objects. The main benefits are the multiple possibilities of individual assignment of responsibilities and contracts, consolidation of developments within an organization, and total control over new developments. It is possible to assign any object or list of objects to a contract or other predefined attributes.

Implementing the Library Definition

The library can be implemented as part of the Solution Manager set-up in the Custom Code Management section step 7.1.

Maintain the CCLM Landscape Setup

Preparing the landscape with the assignment of systems is now part of the Solution Manager set-up in the section Custom Code Management step 4 and 5.

CCLM is designed as a single source of truth, therefore it is absolutely necessary to connect the complete landscape from development system(s) up to the productive system(s) to the Solution Manager where CCLM shall be running. Only with the connection to the complete landscape it will be possible to benefit from the transparency and the features provided by CCLM.
Maintain the SRCSYSTEM Mapping (optional)

In general you will have a clean system with unique source system entries for your custom code objects. This indicates all your custom code objects were created in this specific source system or in your development system. You can find out if this is true, if all the entries in table TADIR within field SRCSYSTEM point to the local System ID. In cases your custom code object belongs to another SRCSYSTEM either you renamed the SYSID of the development system or you made some cross transports. Such data inconsistencies will lead to Custom Code Duplicate entries of an object in CCLM. To avoid this, you will be able to override the SRCSYSTEM value in the custom code objects extractor. Use SE16 to maintain the table 'AGS_CC_MAPSRSYCSYS'. Fill in your Old SYSID in field SID_SRCD_OLD and the correct new System ID in SID_SRC_NEW.

The source system is defining if an object is a named a CustomCodeObject or a CustomCodeDuplicate in CCLM. Therefore the changing of the source system using the table AGS_CC_MAPSRSYCSYS will have an impact in the assignment of objects or duplicates. Please read also the following chapter "Choose a Lead System".

💡 **Hint:** If you get a huge number of CustomCodeDuplicates, there seems to be an issue with a missing SRCSYSTEM mapping. Cross transports or wrong system assignments in TADIR (happens if you rename the system) will lead to this behavior. Please check out if the SRCSYSTEM attribute value for all of your custom code objects fits to the system id of your developments = lead systems.

💡 **Hint:** If you override the source system in the AGS_CC_MAPSRSYCSYS table it will just impact the assignment of object or duplicate. The respective attribute for the source system of this CCLM object will not be overridden, as this shows the original entry from TADIR!

Set up the Collector Jobs

The Set-Up and the scheduling of the collector jobs is now as well part of the Solman-Setup for Custom Code Management step 7.2.

Choose a Lead System

Lead Systems can be defined in the Solman-Setup for Custom Code Management in section 7.2.

You can define one or more lead systems in Solution Manager set-up. You need to define at least one lead system. The lead system is where an object is created (source system), normally the development system. All objects are registered at a technical level in the TADIR table at creation time with the name of the original source system. Your main development system(s) shall be defined as lead system.

💡 **Hint:** For Custom Code Objects the lead system is defining if an object is marked as Custom Code Object or as Custom Code Duplicate. If the source system of an object is defined as lead system the object is marked as Custom Code Object if it is not defined as
lead system it is marked as Custom Code Duplicate. For enhancements and modifications all objects will be recorded of type Custom Code Objects in CCLM.

💡 Hint: To calculate the derived application component of a custom code object the respective references of an object will be checked based on the references of this object in the lead system. This is done as the application component of an object is not system dependent and in order to speed up the process and to avoid to create unnecessary data. If no lead system is marked then the application component will not be calculated. The derived application component of a custom code object (or duplicate) will be stored in the attribute DerivedApplicationComponent.

Typically, each customer landscape will have one development system, one quality system and one production system:

![Typical SAP Landscape](image)

The recommendation is to set the **Lead System** flag for your central Development system(s), as they are considered to be the “master data” systems.

Since objects can be in several systems, for instance, Development, Quality and Production, these systems are linked to the objects as a "Ref2System" entry.

**Remark:** If you have several landscapes, each with its own development system, then you can set each of those development systems as **Lead System**.

### Deletion of Objects

If an object is deleted in one or more systems, then the object collector will no longer find this object in these system(s). Deleted objects will still remain in CCLM, but it will be marked as deleted. This process is triggered by the attribute **ObjectFound** in CCLM. The attribute **ObjectFound** is automatically updated when collector finds the object in a system. If a custom code object is deleted in all systems, this attribute will not longer be updated and the attribute **DeletionFlag** will be set automatically and the **SAPLifecycleStatus** will also be set automatically to "99". If the object is deleted in just one system then the respective "Ref2System" attribute will be displayed with an end date, which is the date it was found the last time in this system by the collector.

**Remark:** The deletion flag will be set by the collector if the date for object found is more than 10 days in the past (field CHECK_REF_DATE in AGS_CC_CUSTOM). This number of days is in sync with the frequency of the collector job, which is set to 7 days with the installation.

### Activate Local Objects (optional)

Local objects are all objects created in development class $TMP. It may not make sense to collect local objects in a development system; developers want to test or try things
and such objects do not belong in the library. Moreover, such local objects will never be transported to a production system.

But if there are local objects in production systems, it means something is developed directly in production, and it makes sense to collect this information and write it to the library to establish transparency with regard to such objects.

Activate this checkbox for a system, if you want to collect local developed objects. The recommendation is to set it up only for production systems (and in limited cases for Quality systems). Highlight the corresponding line entry for the system and click on the button Local Objects.

**Set-up of managed systems**

SAP proposal to set-up CCLM for the different managed systems:

<table>
<thead>
<tr>
<th>system</th>
<th>collection of local object</th>
<th>setting as lead system</th>
<th>collecting objects for CCLM</th>
<th>collecting quality information</th>
<th>collecting usage information</th>
<th>executing reference analysis</th>
<th>executing similarity analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>development</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>quality</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes (if test execution should be monitored)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>production</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes (if not in quality)</td>
<td>yes</td>
<td>yes</td>
<td>yes (if not in quality)</td>
</tr>
</tbody>
</table>

The table above just lists a standard proposal. The set-up might vary depending on customer needs and situation.

**Add or delete customer-specific attributes**

To add customer-specific attributes, you need to have assigned role SAP_CCLM_ALL with authorization object: SM_CC_AUTH with the ID 'SM_CC_LIB' and activity 20 = Change library definition.

- **Start CCLM workcenter**
  
  Library Definition --> Tab attributes

  On the top of the view, the two buttons "Create" and "Delete" should be active.

- **Create:**
  
  To create an attribute, click on the "Create" button and fill the corresponding fields in the pop up:
• **Attribute Identifier**: Name of the attribute. Must start with a prefix for the customer namespace. e.g. "Z".
• Attribute Description: Free text description of the attribute.
• **Attribute Length**: Length of the field in characters.
• **Attribute Type or Reference**: Normally "String"
• **Cardinality**: Use the "1" defaulted value.
• **Reference**: Set this checkbox if you will assign this field to objects.
• **Lower Case (Text)**: Use it if you want the maintained character to be converted to lowercase.
• **Maintainable**: Set this checkbox if you want to maintain this attribute manually.

• **Delete**:
  o To delete an attribute, select it and click on the "Delete" button.
  o You will see a confirmation message "Attribute deleted" on top of the UI.
Define customer specific values

As an example use the predefined search help AGS_CC_SHLP_SEVERITY as a template and replace the default value behind the parameter 'Identifier' with the Customer specific attribute name.

In the next step add the wished search help value in table AGS_CCD_SHLP.

Add your new defined search help to your customer attribute and set the field vhelp = X

Extract data for customer-specific attributes via customer-specific collectors

Once you have defined your custom attributes and know which ones of them should be maintained automatically via a custom collector, then you have to:

1. Define a customer individual RFC function module or a BADI implementation in your managed system(s) to retrieve the data for the custom attributes and pass it back to CCLM.

As a template please use the existing RFC function module /SDF/GET_CCLM_ATTR_VIA_BADI or implement directly the BAdI /SDF/CCLM_ATTR_BADI.

The interface is rather simple to be implemented. The import table _T_OBJECT_SET_ contains all value Custom Code Objects. The output table _T_CCLM_ATTRIBUTES_ will be filled with the corresponding value pairs for each custom code object.
Let’s assume your customer field \_Z_MYFIELD\_ should be filled. In this case take the 3 field contents \textit{PGMID OBJECT OBJ_NAME} from the \_T\_OBJECT\_SET\_ table and feed the structure \texttt{/sdf/cclm\_attr\_4\_badi}.

In the field \texttt{/sdf/cclm\_attr\_4\_badi-CCLM\_ATTR} you enter e.g. \texttt{Z\_MY\_FIELD} and in the \texttt{/sdf/cclm\_attr\_4\_badi-CCLM\_ATTR\_VALUE} you enter any valid value.

If you want to add more than one value pair combination add the corresponding needed entries to the \_T\_CCLM\_ATTRIBUTES\_.

💡 Of course you are also able to feed multiple customer fields at the same time of extraction.

Next step is to propagate the new customer specific function module to the CCLM.

1. Define a new customer field’s extractor job.

For your customer fields you should define your own extractor entry in SolMan\_Setup section Custom Code Management 7.1. Enter your function module in the sector collector configuration.
2. Schedule the custom collector job

The new collector can be scheduled in section 7.2 of the custom code management setup in Solution Manager with the job-name SM_CCL:ATTR_<SID>_<Installation number>.

⚠️ **Attention:** You can decide in section 7.1 if the collector shall be called via RFC in the managed system to collect the data in the managed system, or directly to collect the data in the Solution Manager.

💡 **Hint:** Of course you are also able to replace the extractor name value by an own report, but in this case you have to develop all CCLM updates by your own.

### Define customer specific actions on custom code objects

For this purpose please implement a Filter BADI enhancement called *AGS_CUSTOM_CODE_ACTION*. To learn how it works please follow the example implementation in class AGS_CUSTOM_CODE_ACTIONS. You have to implement 4 interfaces and define a filter value. This filter value is very important to ensure the action will be triggered.

As soon as you implement the BADI, ensure you have a unique filter defined. Assign a customer individual value to the filter AGS_CC_BADI_ACTION_FILTER

Then implement the interfaces:

- **GET_TEXT** = Will prompt you for a popup window with a description of the started action.
- **GET_ID** = Will define the menu entry name
- **EXECUTE_ACTION** = Is the main implementation and process the selected entries
ACTION_PROCESSING = Determines the mode if you want to execute object by object or a mass processing

The "Action" button will only be active, if you have the change and delete authorization for custom code objects.

**Define customer specific GOTO on custom code attribute classes**

For this purpose please implement the Filter BAdI AGS_CUSTOM_CODE_GOTO. To learn how it works please follow the example implementation in IMPL_AGSS_CUSTOM_CODE_GOTO. You have to implement 2 interfaces and define a filter value. This filter value is very important to ensure the action will be triggered. As soon as you implement the BAdI, ensure you have a unique filter defined. Assign a customer individual value to the filter ATTRIB_CLASS.

Then implement the interfaces:
EXECUTE_GOTO = Define what kind of attribute value specific activity will be started with the wished parameters
START = Execute the activity like starting reports or Transaction codes.

**Customer specific CDP determination**

This feature is called **CDP check**. As soon as the custom code objects will be extracted from a managed system and this system is in the role 'lead system' all the related custom code objects with a reference to this system will be send to a generic RFC function called /SDF/CC_CREATOR_CHECK in the managed system. As soon as you maintain the activation customizing by creating an entry CDP in table AGS_CC_CUSTOM the logic for CDP determination will be activated. In the managed system itself you will be able to implement a BAdI implementation for BAdI /SDF/CCLM_ISV_CHECK (please have a look at the template implementation) or create your own RFC Z function module with the same interface as /SDF/CC_CREATOR_CHECK. In this case you have to propagate the name of the Z function module in the customizing value field for the entry CDP.
Possible customizing settings of entry in AGS_CC_CUSTOM

<table>
<thead>
<tr>
<th>Options</th>
<th>CC_KEY</th>
<th>CC_VALUE</th>
<th>Description of behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>No Entry with key CDP: Functionality is completely deactivated</td>
</tr>
<tr>
<td>2</td>
<td>CDP</td>
<td>/SDF/CC_CREATOR_CHECK</td>
<td>CDP check will be executed by calling /SDF/CC_CREATOR_CHECK in managed system. BAdI implementation needed</td>
</tr>
<tr>
<td>3</td>
<td>CDP</td>
<td>Z_RFC_CHECK</td>
<td>CDP check will be executed by calling Z_RFC_CHECK in managed system. RFC function module must be implemented</td>
</tr>
</tbody>
</table>

Maintain Organization Units for the contract/owner maintenance

Use the view maintenance Tcode: SM30 and view AGS_ORGUNIT_CUST to maintain values for the OrgUnits for Contracts. Caution! If you delete any value here it will have no update impact neither on CCLM data nor in the related BW cubes.
CCLM - How to use

⚠️ **Attention:** If you choose "local" in step 7.1 to collect the data for CCLM, the data for CCLM will be collected in steps:

1. Extract data from managed system into Solution Manager BW cubes (starting extractors in step 6.2)
2. Collect data from BW cubes into CCLM tables (defining and starting collectors in step 7.1 and 7.2)
3. Extract data from CCLM into ICI and City Cubes for ICI dashboards and City Modell (starting extractors in step 6.2)

The steps have to be scheduled in this order to show the data in CCLM. This means there will be no data shown in CCLM workcenter before the respective extractor has filled the cubes.

💡 **Hint:** When scheduling extractors and collectors, please make sure that they are scheduled in the right sequence. Especially please make sure that the collector for last usage data is running after the UPL extractor has finished to collect UPL data from the managed system.

---

**Main User Interface**

The application for managing custom code was completely developed using ABAP Web Dynpro and is structured like a work center.

⚠️ **Attention:** The only exemption is the presentation of the "City Model" in the overview screen and the "ICI dashboards" in the dashboard screen. These applications are developed with SAPUI5 technology. These screens will only run with Internet Explorer 10 or higher, Firefox and NWBC 4.0. If the presentation of the city model and
the ICI dashboards are not working when starting the workcenter with transaction SM_WORKCENTER, please try to use transaction CCLM instead.

Starting CCLM with transaction CCLM or with SM_WORKCENTER will open a browser to display the workcenter.

The following sections provide detailed descriptions of all elements and their functions that will help you managing the lifecycle of your custom code.

- Overview
- Library Definition
- Objects
- Quality
- Decommissioning
- Dashboards
- Reports
- Common Tasks
- Related Link

### Overview

The overview provides a view of the City Model of the selected systems. It is possible to choose all systems or select one or more system to be displayed in the city model.

💡 **Hint:** If you want to get the city model in a separate and full screen session just click on the "enlarge" symbol on the top right (see screenshot below).
On the top the overall number of objects are displayed for the selected systems.

With the slider at the bottom of the city model the history of the city can be displayed.

⚠️ **Attention:** The history shows exactly the data from the day/period, which is selected. Especially in case the history is selected for a specific day it might happen that at that specific day the data extractor was not scheduled so no data will be shown for this specific system. In this case a message will be displayed and the system(s) for which no data was extracted for this specific day will be listed, by mouse over (see screenshot below). If data is available on the following days, no action is necessary.

The city itself is build on 3+1 dimension:

1. **Severity:**

Severity is based on the attribute SAPSeverity.
It is possible to collect three different severity levels: Enhancements, Modifications and Custom Code.

These severity levels are automatically collected and stored in the CCLM-attribute SAPSeverity for each object.

SAPSeverity=1 -> Enhancements
SAPSeverity=3 -> Modifications
SAPSeverity=5 -> Custom Code
SAPSeverity 2, 4, 6 are not available.

⚠️ **Attention:** For Modifications (SAPSeverity=3) all modifications will be listed, based on the entries of table SMODILOG in the managed systems: Modifications of custom code objects, of objects from 3rd party and of SAP objects.

💡 **Hint:** If a modification, which is listed in CCLM, is inactive it will be treated as "deleted". This means it gets the lifecycle status "99" and the deletion flag will be set.
For the first run of the object collectors the time range to catch inactive modifications from SMODILOG is set to 365 days. This means only inactive modifications will be collected, which are created between the date the collector runs and 365 days back. This time range can be changed in table AGS_CC_CUSTOM field Mod_Range.

**Hint:** The attribute "Severity" can still be maintained manually by the customer, but this attribute will not be considered to calculate the city model.

### 2. Criticality:

Criticality is based on the attribute SAPCriticality. The criticality can be automatically determined by a collector job for each managed system. The set-up of the critically has to be done in the Solman-Setup for Custom Code Management Section 8.

The criticality level is stored per system in the CCLM table AGS_CC_CRIT. In the CCLM-attribute SAPCriticality the highest criticality level will be stored. The criticality level is distributed from 1 - 5.

1=Low critical  
5=Very critical

**Hint:** The attribute "Criticality" can still be maintained manually by the customer, but this attribute will not be considered to calculate the city model.

**Attention:** Not all object types can be evaluated based on the definition in the set-up. Currently only the object types "PROG" and "TRAN" can be evaluated. All objects which cannot be evaluated appear in the "N/A" column of the criticality dimension in the city model.
3. **Quantity:**

The quantity dimension shows the number of objects for each Severity/Criticality section.

The additional dimension is shown in the two box of the city model dashboard on the bottom of the right hand side. It is possible to select a block in the city model or a severity row and the additional dimension of the selected object block or row is shown on the right on the bottom. In the middle on the right the overall numbers are shown for the additional dimension.

💡 **Hint:** If you want to select another block or row, the block/row that was selected before has to be deselected first.

For the additional dimension it is possible to switch between quality and usage. Just select the respective button in the respective box.

**Usage dimension:**

The usage dimension shows the number of used and unused objects overall and for the selected block/row. The usage data is based on workload statistics and UPL data. From the display of the usage you can directly jump into the decommissioning cockpit by simply click on "Retire unused code".
Quality dimension:

The quality dimension shows the number of good, medium and low objects overall and for the selected block/row. The quality data is based on ATC runs in the selected systems.

An object is rated as:

- low quality, if the ATC runs detect at least one priority 1 message (red message)
- medium quality, if the ATC runs detect at least one priority 2 message (yellow message) and no priority 1 message
- good quality, if the ATC runs detect at least no priority 2 message and no priority 1 message

⚠️ **Attention:** To get the quality data into CCLM an ATC master run must be scheduled on the managed system. This means the new and strategic ATC tool must be implemented in the respective system for all new CCLM installations from SP11 onwards. For existing CCLM installation Code-Inspector results can be considered if the collector configuration for the quality collector in the Solman Setup for Custom Code Management is set to "RFC to Managed System". It is recommended to implement ATC and use the "local" mode in the set-up.

For more information on the implementation of ATC please read:  [http://scn.sap.com/docs/DOC-32791](http://scn.sap.com/docs/DOC-32791)

**Remark:** The information for quality, usage and SAPCriticality is depending on the system. The city model will take care of system dependency. This means if just one
system is selected for the city model, the model shows the data for this specific system. If more than one system will be selected the number of objects in any case which is shown in the model is the addition of the respective number of the selected systems, independent if an object exists in more then one selected system. In fact in this case an object will be counted multiple times!

**Library Definition**

To display objects directly in CCLM, simply select the **Objects** menu on the left side.

To work with the library definition simply select Library Definition menu on the left side.

In this view you can see your active library and you can download and upload you library definition. This function is needed if you would like to export or import your library structure for your customer specific attributes to another custom code library in another Solution Manager system.

⚠️ **Attention:** This function is no longer needed to set-up CCLM. This is now part of Solution Manager set-up.

In the lower section you see an overview on the definition of the available objects and the defined attributes together with the assignment of the attributes to the objects.

---

**Objects**

To display objects directly in CCLM, simply select the **Objects** menu on the left side.
Object classes

CCLM supports different object classes

<table>
<thead>
<tr>
<th>Object class</th>
<th>Description and purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomCodeObjects</td>
<td>Main class. Contains all main objects like programs, tcodes, function groups etc. and sub class of custom code objects with focus on e.g. function modules. Source system for these objects must be marked as a lead system. Enhancements and modifications are always recorded as custom code objects, independent from their source system.</td>
</tr>
<tr>
<td>CustomCodeDuplicates</td>
<td>Special custom code object type. Will be created as soon as an object will be found with a source system which is not marked as a lead system.</td>
</tr>
<tr>
<td>Contracts</td>
<td>Contractors with a relationship to the custom code objects and duplicates. A special role is the contract type CDP representing the creator of the custom code. Further contracts and contract types could be Maintenance; Support, Outsourcing,...</td>
</tr>
<tr>
<td>Owners</td>
<td>Maintained business owner who owns the business responsibility of the code. In best case the requestor. Should not be the developer or the development organization. These owners own the business cases of the developed functionality and are the first contacts for any business related activities.</td>
</tr>
<tr>
<td>System</td>
<td>System active in the CCLM with additional information. CCLM internal purposes.</td>
</tr>
<tr>
<td>Landscape</td>
<td>Will give you an information which Solutions are part of the active extraction. Can be maintained with report RAGS_CC_GET_LANDSCAPES</td>
</tr>
<tr>
<td>Analysis Tool</td>
<td>Information about the current CCLM feature pack. CCLM internal purposes</td>
</tr>
</tbody>
</table>

Custom Code Object selection and details

With SP12 all customer objects from namespaces, which are defined as customer namespaces, will be collected, if the objects are listed in TADIR. These means, that all object which are defined in TADIR from customer namespace will be collected automatically. In fact there are now more object types collected in CCLM from SP12 than before.
The objects collected are displayed as a list in an upper view. There, in addition to the object ID, the object class, technical name, creation date in the library, last change date and last user that changed the object are displayed. There is a filter area above the list that can be expanded. There are also filtering options for every column in the list. An additional search function enables the search for special attributes that can be assigned to the objects. This search function may be rather time-consuming depending on the number of objects and the hit list. We recommend setting a filter (customizing setting) to 200 or 500 hits to ensure that the result can be displayed quickly.

**Standard attributes**

Which attributes have been set or maintained by you is displayed in the OBJECT DETAILS for each object in the list below the object list in figure. The standard attributes for a customer object either come from the TADIR table or are defined and non-maintainable attributes from the definition that are automatically filled, for example the attribute Ref_2_System, where the system in which the object with the respective object ID was found is listed.

Standard attributes are read only and cannot be maintained manually. Updates to these fields happen always automatically.

**Maintainable attributes**

Optionally, there is a tab that stores maintainable attributes. Here you have the option of creating, changing or deleting changeable attributes. Which attributes of an object can be changed or created is defined in the library definition. Customer specific attributes will be part of the maintainable attributes.

**Filter objects by**

To limit the number of displayed objects you can expand the tray in the right upper corner.

You will have now the possibility to filter the objects by
In all fields wildcards can be used. The Create Date or Change date represents the dates when the custom code objects was introduced in CCLM library or received a change. The change date will be adapted as soon as the object will be found again in an attached system.

💡 A speed filter is available to filter the objects list directly in the ALV list.

### Search objects with attributes

Expanding this tray you will have the possibility to filter any kind of assigned ATTRIBUTES and VALUES.

E.g. If you want to see only Custom Code Objects with a "Devclass = ZABC" Just maintain the values.

⚠️ **Attention:** The selection of possible attributes depends on the selected Object Class in the 'Filter by' tray.

### Object type PROG

With SP12 includes are collected for CCLM with object type PROG as well. In order to identify includes the attribute "Program Type" will be filled with the following values:

1 = Executable Program  
I = Include  
M= Module Pool  
S= Subroutine Pool

Please implement note 2116608.

⚠️ **Attention:** For includes of SAPSeverity = 1 or = 3, this "Program Type" will currently not be filled.

### Owner Concept

There are numerous functions for the displayed objects, such as the assignment of references and maintenance of attributes. You can also create, change and delete the objects **Responsible person/Owner**.

#### Business Owner (assigned directly)

The main purpose of the **Owner** concept when assigned directly to an object is to identify and someone responsible from the business area. This owner is normally the person who presented the business need and created the business requirement.

💡 **Keep in mind:** The developer created the custom code is not the business owner. He is the creator only and for the most part available on project base only.
Examples for owners are real life representatives from an internal organization like Financial, Logistics, Sales, Distribution, etc. You should have only one owner assigned to each custom code object (although multiple assignations are possible).

**Organization Owner (assigned indirectly via a contract)**

An owner could also be created to represent an organization responsible for an object at a technical level, but they should not be assigned directly to the object, but rather indirectly, by assigning the objects to a contract and then the contract to an owner.

**Owner Creation**

To create an owner, select the button **Create Owner** in the overview and fill the appropriate data. Maintain at least the mandatory fields.

![Add Object Owner](image)

You should assign a real SAP user from the Solution Manager system. Maintaining also some contact information will help you in emergency cases when there is the need to contact the owner to obtain business related information about the custom code object.

**Contract Concept**

A contract is means to group multiple objects that have similar characteristics and assign them to an organization responsible for them.

In most organizations you have a separation between development, maintenance & support and sometimes also outsourcing teams. However, normally only one is responsible to maintain the object at a technical level.

The main purposes of having contracts is keeping transparency about the current activities on custom code objects and assigning technical responsibilities on organizations rather than on real life people.

Thus, objects should be assigned to a contract rather than to an owner for technical purposes. There are two optional owners for contracts: Responsible Owner and Personal Owner. The first one represents the person closest to the object (i.e. developer) and
normally who created it. The second one represents the overall responsible (i.e. manager) for a big group of objects.

**Contract Creation**

To create a contract, select the button **Create Contract** in the overview and fill the appropriate data. Maintain at least the mandatory fields.

**Contracts in role of a software provider**

There is a feature available called CDP related contractors. A CDP will be an organization or a department within the customer organization or also an outsourced independent software vendor (ISV). If you outsource your custom code development to e.g. SAP Custom Development, you should create a special contract type called 'CDP' for this purpose. SAP Custom Development as a CDP will be generated automatically in your customizing as soon this feature will be activated. The contract type "CDP" will be a reserved name and lead to addition CCLM activities.

**Change Owner/Contract description**

It is always possible to change the description of owners or contracts. This can be used to make minor changes as long as the organizational elements will not be changed. If you want to reassign 100 custom code objects from person A to person B, then you have to do a mass maintenance assignment.

This change functionality is designed to change the name of person A without changing all custom code relations.

**Delete Owner/Contract**

To delete an owner or contract, select the object in the corresponding line and click on the button 'Delete'. This will remove the owner/contract from the master data and remove all assignment to any custom code objects.

**Add reference**

Assignation of individual or multiple objects to a contract or owner should be done in order to be able to manage Custom Code efficiently. To do so, use the **Add a Reference button**, pick the corresponding type (contract or owner) and then select the contract or owner from the list.
Once you have done it, the Attribute Class “Ref_2_Contract” or “Ref_2_Owner” is displayed for each object in the tab “Maintainable Attributes”. You can then filter object by contract, owner or by both (AND condition).
**Maintain attributes**

It is possible to manually maintain attributes for custom code objects. Some examples of the standard attributes are: criticality, severity level and comments. To do so, use the “Maintain Attributes” button, pick the corresponding attribute type and then select the corresponding value.

**Mass Maintenance of objects**

Filter, Search and multiple select and maintain attributes or references

The list is designed so that you can select multiple objects and thus perform a sort of mass maintenance. One example would be the assignment of a maintenance contract to all objects of a particular development class. To do so, you must set a filter for the respective development class in the list. In this way, you can select all objects in the first column before the object ID at once. Then you can click the ADD REFERENCE button, select the desired contract in a popup and confirm your selection.

**Quality**

To get an overview on the available ABAP Test Cockpit (ATC) runs, simply select Quality menu on the left side.

⚠️ **Attention:** In order to get results in this screen ATC must be implemented in the managed system(s) and a master run has to be defined to analysis the objects. This is only working for ATC ist is not working based only on code-inspector results.

On the top you can select the period, a specific ATC run and a specific check variant.

In the middle please select the system you would like to see the results. Which will then be displayed in a chart on the right. The different lines represent the number of prio 1, prio 2 and prio 3 messages detected by ATC in the different runs.
More details for each run are displayed in the table below the graph.

You can select a specific ATC run to display the detailed result for the single objects.

**Decommissioning**

The Decommissioning Cockpit supports the decommissioning strategy. It supports the deletion of custom code objects in your landscape.

If an object is deleted in the complete landscape it will still remain in CCLM, but the attribute "DeletionFlag" for this object will be set to "X" and the "SAPLifecycleStatus" will be set to "99".

The procedure is as follows and the respective attributes will be set by the object collector.
Any time the object will be found in any system the attribute "ObjectFound" will be set to "X" and the change-date for this object will be updated.
If an object will be found for the first time in a system the system will be recorded in the "Ref_2_System" attribute together with the date when it was found first. This date will never change.

|--------------|-------------------------------|------------|----------|

Any time the object will be found in a system the change date of the respective "Ref_2_System" attribute will be updated.
If an object is deleted in a system the "Ref_2_System" attribute will receive an end date if the last date when it was found is >= 10 days back. The decision will be made by checking the 10 days against the change-date of the "Ref_2System" attribute. This means it takes at least 10 days before the end date will appear in the "Ref_2_System" attribute.

<table>
<thead>
<tr>
<th>Ref_2_System</th>
<th>ST_000270882 (since 28.04.2014)</th>
<th>01.04.2014</th>
<th>15:58:06</th>
</tr>
</thead>
</table>

If an object will no longer be found in any of the connected system, the collector job will check any time the change-date of the attribute "ObjectFound". If the change date of this attribute is >= 10 days back the attribute "ObjectFound" will be set to " " and the DeletionFlag" to "X" and the "SAPLifecycleStatus" to "99"

**Remark:** The deletion flag will be set by the collector if the date for object found is more than 10 days in the past (field CHECK_REF_DATE in AGS_CC_CUSTOM). This number of days is synchronized with the frequency of the collector job, which is set to 7 days with the installation. At any time the number of days to look back must be greater then the number of days for the frequency of the collector job for any system.

**Decommissioning Cockpit**

The decommissioning cockpit provides an overview on the current decommissioning analysis.
On the top the number of analysis are shown in four blocks with regards to their current status. In each block it is possible to click on the list and directly jump into the list view of the selected analysis. In the "New Analysis" block it is possible to change the screen to start a new analysis.
In the middle of the cockpit an overview is provided of the number of objects in each lifecycle status. You can select the objects by entering "Object Selection Criteria" and press the refresh button.
On the bottom an overview is shown on the usage of all objects from all systems for a specific number of days in the past.
Attention: The two status "Active in Non-Production" and "Under Development" are not available in this version. Once an object will be found only in the system which is the source system of this object the status will be set to "New Object". If it will be found in any other system, which is not the source system, it will be set to "Active in Production".

Create New Analysis

Attention: In order to use the decommissioning cockpit for your decommissioning project it is absolutely necessary, that workload statistics and UPL data is available. Without these two data source usage data cannot be calculated and the decommissioning cockpit cannot be used to decommission objects based on usage information. Please make sure that workload statistics is available and UPL is running on the different systems.

Attention: A decommissioning analysis can only be created for objects of SAPSeverity = 5. All other objects with a different SAPSeverity level will be excluded from the analysis automatically, although they might fulfill the selection criteria.

Selecting "Create New Analysis" on the top of the decommissioning cockpit in the "New Analysis" block will open the screen to create a new analysis.

Beside some generic information like name and description the selection of objects have to be entered. There are three standard use cases:

1. Objects which are not used in specific time frame in specific system(s)
2. Objects which are not active in specific system(s)
3. Objects which are similar to SAP objects (clones)

If none of these standard selections apply to your project settings choose "Other" and select the objects via the assigned attributes.
In order to be able to directly send an e-mail to the analysis owner, when an object was used during waiting period the owner has to be maintained as an owner in CCLM and the respective SAP User in the Solution Manager is maintained in the owner data as well. The e-mail address will be taken out of the SAP User.

Save and start the analysis by pressing the button on the top on the right hand side.

⚠️ **Attention:** What does it mean "unused objects" in an analysis period? An objects fulfills the criteria of an "unused object", if:

1. the last usage date for any statistical system is before the start date of the analysis period
2. there is no last usage date recorded for any statistical system so far, but UPL or workloadstatistic is able to record the usage for this object type
3. there is no last usage date recorded for any statistical system so far and UPL or workloadstatistic is **not** able to record the usage for this object type (e.g. WDYN, TABL, ...)

⚠️ **Attention:** This means that the result list of the analysis has to be checked carefully, which case is relevant for the specific object, before decommissioning the object. Especially case three needs to get attention.

**Role of “Lead Systems” and “Statistical Systems” entries in settings for the selection of objects.**

Entry field "Lead Systems": It is recommended to select lead system(s) (as defined in the set-up) or at least development system(s), which are connected to CCLM. Although this entry field is named "Lead System" any system connected to CCLM can be selected.

Entry field "Statistical Systems": Any system, which is connected to CCLM can be selected.

1. Selection of not used objects in statistical systems:
   An object will be classified as not used, if
a. (existence check) the object was found by CCLM object collector in at least one of the statistical systems (at least one Ref2System entry for a statistical system) and if the object was found in at least one of the lead systems (at least one Ref2System entry for a lead system) and if
b. (usage check) there exist no usage data (last usage date) for this object in CCLM for any of the statistical systems.

2. Selection of not active objects in statistical systems:
   An object will be classified as not active, if:
   a. (existence check) the object was found by CCLM object collector in at least one of the lead systems (at least one Ref2System entry for a lead system)
   b. (non existence check) the object was not found by CCLM object collector in any statistical system (no Ref2System entry for any statistical system).

3. Selection of similar objects:
   An object will be classified as a similar object (clone), if
   a. (existence check) the object was found by CCLM object collector in at least one statistical system or lead system (at least one Ref2System entry for a lead system or a statistical system) and if
   b. (similarity check) the object was found by CCLM similarity collector in at least one of the lead systems or statistical systems.

4. Selection "Other":
   An object will be classified for this selection, if
   a. (existence check) the object was found by CCLM object collector in at least one statistical system or lead system (at least one Ref2System entry for a lead system or a statistical system).

This is the basic selection of the objects for the decommissioning analysis. Beside these basic selections based on lead and statistical system entries the additional criteria will be checked if applicable (e.g. analysis period, similarity degree, creation date, ...).

**Analysis List**

Selecting analysis lists in one of the blocks on the top of the decommissioning cockpit will open the list view of the analysis.

On the top the analysis list is presented. In the status column the list can be filtered by the respective status. Select a analysis which you would like to work with. The "Object Statistic" provides an overview on the current status of the selected analysis.

On the bottom the assigned object to the selected analysis will be listed by pressing "Get Object List". You can filter the objects of the selected analysis by entering "Object Selection Criteria" on top of the list and press "Get Object List" again.
It is now possible to work with the objects by selecting the objects. The lifecycle status of all objects, which are part of a decommissioning analysis, will be set to "Recommended for Decommissioning".

Before working with these objects the status of the object has to be changed to "Identify and Waiting".

⚠️ **Attention:** If an object appears in more than one analysis it will be listed in all analysis. If in one analysis the lifecycle status will be changed to any status higher then "Recommended for Decommissioning" from this time on the object can only be maintained in this analysis. In all other analysis this object will be greyed out in the list.

You can select the objects in the list and group them to different packages and assign a package owner with button "Email Package". It is possible to directly send an e-mail to the package owner. Therefore the owner has to be maintained as an owner in CCLM and the respective SAP User in the Solution Manager is maintained in the owner data as well. The e-mail address will be taken out of the SAP User.
You can also select objects from the list and group the objects for a backup transport, which can be created with the button "Backup Transport". As soon as the backup transport is created the number of the backup transport will be recorded in CCLM and the lifecycle status of the objects in this transport will be set to "Backed Up".

In the same way a deletion transport can be created. The number of the deletion transport will be recorded in CCLM and the lifecycle status will change to deleting.

⚠️ **Attention:** A backup transport must be created, if in the settings of the analysis it is marked that a backup transport is mandatory. If the backup transport is not created from Solution Manager, then the status of the objects must be set manually to "backed-up" before a deletion transport can be created from Solution Manager.

⚠️ **Attention:** It will not be checked if for the backup transport and the deletion transport dependent objects are part of the transport as well. Therefore only for PROG and TRAN a deletion transport can be created.

The back-up and the deletion transport will be created in the system which is listed in the column "Dev Sys" in the object list of the selected decommissioning analysis. Please make sure that all selected objects of a transport have the same "Dev Sys" listed. The column "Dev Sys" is filled with the respective source system of each object.

The waiting time was defined for the whole analysis, but it is possible to change the waiting time for each individual object. The waiting time can be changed for an individual object by pressing the button "Change Wait Time".

If an object should no longer be part of the analysis the object can be rejected and the lifecycle status will change back to the status before it was selected.

All attributes of each object can be shown in a pop-up with button "Details".

All attributes can be maintained for the selected object(s) with the button "Maintain Attributes" in the same way like in the object view of CCLM.

The lifecycle status of the objects can also be set manually.

The "Used Status" column indicates if an object was used in the mean time. This means if it was used after it was selected for the analysis and before the waiting time was over. A message to the analysis owner will be send if the respective selection was marked when creating the analysis.

**Dashboards**

⚠️ **Attention:** The dashboards for CCLM delivered until SP11 have been replaced by new and more powerful dashboards from SP12 onwards. The dashboards provided until SP11 and the respective queries cannot be used anymore.

⚠️ **Attention:** These dashboards are developed with SAPUI5 technology. These dashboards will only run with Internet Explorer 10 or higher, Firefox and NWBC 4.0. If the presentation of ICI dashboards is not working when starting the workcenter with transaction SM_WORKCENTER, please try to use transaction CCLM instead.

Starting the ICI (interactive continuous improvement) dashboards by selecting "Dashboards" in the control panel on the left will lead to an overview of this seven KPIs.

CCLM - How to setup
All relevant numbers are calculated from CCLM data, by extracting the relevant CCLM data into specific ICI cubes in the Solution Manager BW.

⚠️ **Attention:** Depending on the extractor runs there might be a delay between the data in CCLM and the data shown by the ICI dashboards.

⚠️ **Remark:** If the "Dashboard" button is not available in the control panel, please activate the field "BI_ANALYSIS" in table AGS_CC_CUSTOM. (See also chapter "Special customizing of CCLM" below)

There are seven KPIs, which will be delivered in standard. These KPIs represent information of the 3+1 dimensions of the city:

- Total number of Custom Code Objects
- Total number of Custom Code Objects with low quality (red messages)
- Total number of not used Custom Code Objects
- Percentage of Custom Code Objects with Criticality Level 4 and 5
- Total number of Custom Code Objects with SAPSeverity Level 1 - enhancements
- Total number of Custom Code Objects with SAPSeverity Level 3 - modifications
- Total number of Custom Code Objects with SAPSeverity Level 5 - independent custom code objects

### Business Value

This overview is showing the numbers for all systems. With a click on a specific KPI more information will be displayed.

### Key Performance Indicator

Selecting "Key Performance Indicators" will show all relevant information for the seven KPIs. Select "Custom Code Management" in the KPI group to show the relevant KPIs for Custom Code Management.
The following information is displayed for all seven KPIs for all systems:

- Status: Overall status of the KPI
- KPI: Description of the KPI
- Trend: Changes/Trend of KPI
- Baseline
- Achievement: Current achievement
- Target: Target number
- Benchmark
- KPI current: current value of KPI
- Monthly improvement

At the bottom best and worst KPI will be named with regards to the current achievement and the monthly improvement.

The following numbers are calculated and displayed in the ICI dashboards:

The example is based on the KPI "Total number of Custom Code Objects"

**Baseline**

The baseline is the starting point for each KPI. It can be calculated automatically by the system as the first measurement of the KPI or it can be set manually by the user (within the ICI configuration). This might be the case if for example the first measurement by the system includes not all systems then the baseline can be set manually to the overall number for all systems afterwards. The baseline might also reflect the number which should not be exceeded.

Example: baseline = 60000 objects

**Target**

The target is the number, which should be achieved at the end of a specific time frame. Therefore a **start date** and a **target end date** must be defined.

Example: target = 40000 objects starting March 1st with target end date December 31st.
**Benchmark**

The benchmark is provided by SAP for all KPIs. It is calculated from SAP's experience out of several customer projects.

Example: benchmark = 12000 objects

**KPI current (measurement)**

KPI current represents the current value of the KPI in the system.

Example: March = 60000 objects (baseline), April = 59500 objects, May = 57350 objects, June = 58100 objects, July = 56100 objects, August = 47020 objects current day August 25th.

**Achievement (value and %)**

The achievement will be calculated by the system. 100% achievement will be calculated if the target value is reached. Therefore the difference between the baseline and the target is calculated and set to 100%. For the current achievement the difference between the baseline and the current KPI value is calculated. This result will then be set into relation to the 100% achievement.

Example: 100% achievement = baseline-target = 60000 - 40000 = 20000 objects

current achievement = baseline - KPI current:
March: 60000 - 60000 = 0; April: 60000 - 59500 = 500; May: 60000 - 57350 = 2650;
June: 60000 - 58100 = 1900;
July: 60000 - 56100 = 3900; August: 60000 - 47020 = 12980
achievement (%) = current achievement month/target achievement * 100; current achievement in month in relation to target achievement (target achievement = 100%)
March: 0/20000 *100 = 0; April: 500/20000 * 100 = 2,5; May: 2650/20000 * 100 = 13,25;
June: 1900/20000 * 100 = 9,5;
July: 3900/20000 * 100 = 19,5; August 12980/20000 * 100 = 64,9

**Set-Value (value and %)**

Based on the baseline and the target a set-value will be calculated for each month. Therefore the number of month between the starting date and the target end date will be calculated. The 100% achievement (delta baseline - target) will be divided by the number of calculated months. Following a linear approach this is the target achievement, which should be reached each month. The set-value is now the baseline changed by the calculated target achievement for the first month and set-value of month before changed by the calculated target achievement for each following month.

Example: start date: March 1st; target end date: Dec. 31st. Number of month March to December: 10 month

100% achievement: 20000 objects. Set-value for each month: 20000 / 10 = 2000 objects

Set-value for each month = baseline(set-value of month before) - set value per month:
March: 60000-2000 = 58000; April: 58000 - 2000 = 56000; May: 54000; June: 52000

... November: 42000; December: 40000

Set value (%) = set-value month/baseline * 100; set-value month in relation to the baseline (baseline=100%)
March: 58000/60000 * 100 = 96,67; April: 56000/60000 * 100 = 93,33; ... Nov: 42000/60000 *100 = 70; Dec: 40000/60000 * 100 = 66,67

**Plan/Actual**

This is the relation between plan (set-value of each month) and actual (current KPI value). It will be calculated by setting plan to 100% and calculating percentage for
actual in relation to the plan. From this result 100 is subtracted as we would like to show the difference to 100%.

Though higher the value though greater the distance to the set-value of each month. Is the number positive the current KPI value is still above the set-value, it is negative if the current KPI value is below the set-value.

Example: Plan/Actual = (current KPI/set value month *100) - 100
March: (60000/58000 * 100) -100 = 3,45; April: (59500/56000 * 100) -100 = 6,25;
May: (57350/54000 * 100) -100= 6,20
June: (58100/52000 * 100) - 100 = 11,73; July: (56100/50000 * 100) -100 = 12,2;
August: (47020/48000 * 100) -100 = -2,04

Monthly Improvement

The monthly improvement shows the improvement of the current month in relation to the last month. The KPI value of the last month will be set to 100% and the current KPI value of the current month will be calculated with relation to the 100%. From this result 100 is subtracted as we would like to show the difference to 100%.

Though higher the value though greater the difference to the KPI value of the month before.

Is the number positive the current KPI value is above the KPI value of the month before, it is negative if the current KPI value is below value of the month before.

Example: Monthly Improvement = (KPI value current month/KPI value last month * 100) - 100
March: no result, as there is now value for last month; April: (59500/60000 * 100) -100 = -0,83; May: (57350/59500 * 100) -100 = -3,61;
June: (58100/57350 * 100) -100 = 1,31; July: (56100/58100 * 100) -100= -3,44;
August: (47020/56100 * 100) -100= -16,19

Input Measures

Selecting one of the KPIs with a click will lead to detailed historical information. Select "Input Measures" to see the historical details on the overall numbers. On the left the overall number for each system with historic data is displayed. You can select the granularity of the historical data (weeks, month, days) on the top on the right.

On the right a system can be selected as well to get more details on the respective number for each severity level. This number can be drilled down to the specific object types. The object types, which will be displayed can be selected individually. The settings for the object types can be done by selecting the "iCI Configuration" button on the left.
Quick View

Selecting one of the KPIs with a click will lead to detailed historical information. Select "Quick View" to see the historical details on the achievement, plan/actual, KPI value and monthly improvement. You can select the granularity of the historical data (weeks, month, days) on the top on the right.

BW Reporting

This section replaces the "Reports" section from SP14 onwards. It is now possible to define specific reports on CCLM data without using the Web Application Designer. The reports will access the generic BW cube for CCLM (0SM_CCL). In this info cube data from CCLM will be transferred via the respective extractor (see chapter CCLM Set Up above). Most of the standard attributes are already available in the cube. In addition there is the possibility to map additional standard attributes and customer specific attributes to 30 free fields in the cube (fields: 0SM_CCLMxx). After the mapping these fields will be filled with the values of the mapped fields from CCLM.

⚠️ **Attention:** Please map the fields before you start the CCLM extractor for this cube. After the fields are mapped please do not change existing mappings, as the historical data will not be changed. Of course if there are still free fields they can be mapped to other fields at every time. From the point in time the mapping was done, with the next run of the extractor the newly mapped field will be filled.

Under the section "Add New Attribute To CCLM" the free fields of the cube can be mapped to CCLM fields. Just click on "Attributes Mapping to CCLM Infocube".

A new window will pop-up where the existing standard or customer attributes can be mapped to the free attributes in the cube.
Select the section "CCLM Reporting" "Load data from BW" in the CCLM workcenter to define the report and to start the analysis.

After selecting this section the available fields will be shown in the "Navigation Block". Each existing field can be selected to be part of the analysis either in a row or in a column. In addition a filter can be defined for each selected field.

After defining the report, the result of the report will be immediately shown below in the section "CCLM BW Reporting".
**Common tasks**

**Call Ad-hoc reporting**
A comprehensive reporting function known as Ad-hoc Reporting is available. You can find this application as a link under **Common Tasks**. These are at a glance the report areas:

**General Area:**

The library information with key, ID and version must always be filled to get results. By default the objects of type Custom Code Objects are selected. The other option is to display Custom Code Duplicates.

Select what is desired to evaluate: **All Objects** (default), **New Objects**, **Used Objects** or **Unused Objects** (=Objects with no usage information).

**System Selection for Transaction Data Area:**

In this area it is possible to select a system or group of systems for the transaction data (usage, version and quality), which is the only data for which system information is saved. For system ID it is recommended to use the F4 help, as the installation number is required as well.

**Run Options - Period Selection:**

If you select usage or quality a new section will appear to select periods for this transactional data.
Run Options:

As a specific run option, you can select by use, version, quality, similarity or just a simple object inventory. The most common scenarios are the use of objects, and the object inventory. We recommend that you save your frequently used settings as variants of the report. This allows you to start the reporting quickly and get results faster.

<table>
<thead>
<tr>
<th>Run Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Inventory</td>
</tr>
<tr>
<td>Usage</td>
</tr>
<tr>
<td><strong>Used/Unused Objects</strong></td>
</tr>
<tr>
<td>Unused Objects</td>
</tr>
<tr>
<td>Used Objects</td>
</tr>
<tr>
<td><strong>Usage Display Options</strong></td>
</tr>
<tr>
<td>Show no usage details</td>
</tr>
<tr>
<td>Show summarized usage count</td>
</tr>
<tr>
<td>Show last used date per object</td>
</tr>
</tbody>
</table>

Attention: If you want to select used objects, there are three sub-options for this selection.

The first two options "show no usage details" and "show summarized usage count" are only available if CCLM was already active before SP11. These options allow displaying the old usage information from table AGS_CC_USAGE with the results before SP11. This table will no longer be valid from SP11 onwards and will no longer be filled with new usage data after SP11. This means, that the results of this selection is based on data which is no longer updated, so these result do not reflect the current status of CCLM used objects.

From SP11 onwards this table was replaced by AG_CC_LAST_USG. This new table only holds the entries for the date of last usage of each object per system. This table will be displayed choosing the option "show last used date per object". Only this option shows the current status of used objects.

The migration from the old table to the new table looks as follows: The data respectively the last usage date will be calculated out of the existing UPL history as part of the CCLM set-up in step 1 and in addition the Last Usage Collector checks within its execution if the data from the old table has any entry, which was not yet migrated. The second part is needed, as transactions are not recorded by UPL. This new table only holds the entries for the date of last usage of each object per system. This table will be displayed choosing the option "show last used date per object".

Attention: If you want to select unused objects, please be aware that there is a preselection of object types already predefined in the report.
This means that, with this selection the report focuses only on object types where it is possible to collect usage information (see chapter "Collecting Usage" above). To make the list more readable we exclude object types where usage cannot be collected, like for example SMIM or WAPA. As a result please be aware that the number of used and unused objects on the lists created with ad-hoc reporting do not match the overall object number in CCLM. To get the overall object number of CCLM simply create a complete inventory list in ad-hoc reporting. Please implement note 2220716 (SP14).

Filter and Attributes

In this area concrete, multiple filters can be specified. Filters and attributes allow you to control the search for particular data as well as the output with the number of columns. You can filter by any particular attribute or combination of attributes. This also applies to references to the objects OWNER and CONTRACT. Here the columns the objects PERSON RESPONSIBLE and CONTRACT. Also, additional columns (additional attributes) can be added to the ALV display before running the report.

Related Links

⚠️ Attention: Please be aware that you might need additional authorization to call CDMC, Scope and Effort Analyzer and Custom Code Roadmap.

Custom Code Transparency

CDMC

Many SAP customers modify or enhance their SAP standard software. For example, they may create company-specific reports or custom (externally or internally developed) add-ons. Using custom code and modifications enables companies to adapt business processes to market requirements. The result of this natural development is a multitude of customer objects and modified SAP standard objects in circulation. However, requirements change so quickly that many customer objects and changes to SAP standard objects quickly become obsolete. Experience shows that after just a few years, a third of custom code is either no longer in use or has been modified, and not only due to fast-changing requirements but also because new versions of the SAP standard software contain objects that render custom code useless. This can lead to unnecessarily high maintenance costs, which in turn cause high operating costs. For customers, it can be difficult to keep up with the pace of modifications to the standard system and produce custom code. Numerous changes and customer objects also raise the cost of upgrades and importing Support Packages. Before a system can be upgraded to a newer SAP version or a Support Package can be imported, all custom code must be checked and manually adjusted. Then all objects have to be checked one-by-one to ensure that they do not cause problems in the new context.

Using the Custom Development Management Cockpit (CDMC), you can determine how custom code is used (based on the call statistics provided by the system) and which customer-specific developments are obsolete. The CDMD then evaluates the effects of an upgrade or a Support Package installation on custom code. The business process documentation for custom code is also determined (maintenance using transaction SOLAR02). CDMC also supports the project or release manager in evaluating risk by analyzing objects from transport orders before importing them into the production system by means of the Change and transport system analysis.

You must ensure that planned changes are implemented in line with business requirements. CDMC simplifies upgrade projects by reducing the amount of obsolete custom code. Upgrades can be performed more quickly because you only have to modify
your custom code if it is absolutely necessary. CDMC supports project planning by enabling early estimation of the costs of adjusting object types that are required for a more current version. You profit from better and more reliable planning and shorter project runtimes, and thus reduced costs. Another advantage is that the objects in transport orders are analyzed before being imported into the target system.

**Custom Code Quality**

**ABAP Code Inspector**

**ATC Results**

⚠️ **Attention**: This link might still be visible in an installation of SP12+, but it is no longer needed as this and more information will be presented in the quality menu as described above.

**ATC Open Exemptions**

⚠️ **Attention**: This link might still be visible in an installation of SP12+, but it is no longer needed as this and more information will be presented in the quality menu as described above.

ATC results are integrated into CCLM. The ATC results and exemptions can be viewed from the CCLM menu 'Custom Code Quality'.

---

**Custom Code Lifecycle Management**

- **Overview**
  - **Settings**
    - **Systems**
      - My Systems - empty (0)
      - All Systems - All Solutions (31)
    - **Active Systems**
      - Nothing is active for (0)
      - All Active Systems (4)

- **Library**
  - **Active Library**
    - Custom Code Library (1)
  - **Inactive Libraries**
    - Test Libraries (0)

- **Related Links**
  - **Custom Code Transparency**
    - Custom Code Analysis
    - CDMC
  - **Custom Code Quality**
    - ABAP Code Inspector
    - ATC Results
    - ATC Open Exemptions
  - **Custom Code Documentation**
    - Custom Code Roadmap
    - CCLM Best Practice
    - CCLM Online Help

CCLM - How to setup
Please ensure that the role 'SAP_SMWORK_CCLM' is assigned to the user. (This option is valid from SP>09).

Besides the SAP standard code quality tools like SAP Code Inspector or SAP ATC there exists customer individual solutions or solutions from partner and 3rd party. (e.g. Smartshift, Virtual Forge,...)

CCLM is also able to take over results and relations to these external products.

Prerequisites:

- External quality tool offers an interface to extract data
- New CCLM attributes will be maintained
- New collector will be developed and integrated. See chapter "New Customer specific Attributes"
- Implementation of BAdI to activate actions on object lists
- Adoption of BW dashboards Please get in contact with corresponding external partner to get the CCLM enhancement.
Criticality (Solution Manager Set-up -> Custom Code Management -> Step 8)

Criticality is one of the dimensions of the custom code city model. In order to get a complete view of the city the criticality should be defined.

As the city model is only available from SP12 onwards there is no need to set-up the criticality in SP11 already.

To set-up criticality the step 8 with the respective sub steps have to be executed in the Solution Manger Set-Up in the section Custom Code Management.

The criticality is calculated based on 5 data pools:

- **Usage**: Number of end users using this business transaction. The highest and lowest execution values are relevant.
- **Frequency**: Frequency of business transaction and program execution. The most and the least used transactions and programs are relevant.
- **Runtime**: Overall time to execute the business program or business transaction.
- **Data Growth**: Increase in the volume of data related to business processes in the context of transactions and reports.
- **Table Utilization**: Overall number of times a table is used by transactions and programs.

Although it is called "Business Criticality" in the set-up, the calculation is based on technical measurable facts, which are available in the systems.

⚠️ **Attention**: Not all object types can be evaluated based on the listed data pools. Currently only the object types "PROG" and "TRAN" can be evaluated. All objects which cannot be evaluated appear in the "N/A" column of the criticality dimension in the city model.

For usage, frequency and runtime the data will be extracted out of the workload statistics, for data growth and table utilization the data will be extracted from Data Volume Management (DVM).

**Select the data pools (step “Analysis”)**

In order to calculate the criticality for specific object types the relevant extractors have to be scheduled and sufficient data must be available. Sufficient data means at least one month of data must be available. You can define the period, which should be used to collect the data from 1 month up to 1 year.

You don’t need to select all data pools for the calculation, but at least two pools have to be selected. For each selected pool a criticality rating can be specified. This criticality rating refers to how important you consider each data pool and is used to determine the default weighting factor of each pool:
• Low = rating factor 1
• Medium = rating factor 2
• High = rating factor 3

For the selected data pools, specify the upper and lower thresholds of which objects you want to rate as critical in the analysis. For example, if you set both the upper and lower limits to 10 for the Frequency pool, the least used 10% and most used 10% of objects are rated as critical.

You can check for each selected pool if necessary data is available by pressing the button "Check Data". The LED icons indicate whether sufficient data exists for each data pool on the selected systems.

The following statuses are available:

• **Green**: Sufficient data exists and you can proceed to the next step.
• **Yellow**: Partial data is available for the selected period. For example, if you selected a period of 3 months for the data pool but only 2 months of data is available.
• **Grey**: There is no available data, although the extractor is active. This might be because the extractor is scheduled but has not yet collected data. For example, if the extractor is scheduled to run every week but was only activated 2 days ago.
• **Red**: You have not activated the relevant extractor. Go back to the Activate Extractors step.

The extractor for the workload statistics should be activated in the basic configuration and the extractor for DVM can be activated in solman_setup section Data Volume Management step 8 "Activate DVM Extractors".

**Define weightage for each data pool (step “Apply Weightage”)**

To calculate the criticality it is necessary to check in which data pool an object is rated as critical. For the calculation all possible intersections of all possible combinations between the selected data pools will be considered. The means the overall criticality of an object will be calculated in relation in how many intersections this object is included.

⚠️ **Attention:** If a combination is greyed out, this means that there is currently insufficient data available for one or more of the pools in this combination.

For each intersection a specific weighting factor can be defined. The default factor is based on the rating of the data pools itself.

Remember: Each criticality rating of a data pool corresponds to a numerical value, that is, the Low, Medium, and High criticality ratings are converted into a weighting factor of 1, 2, and 3 respectively. The default weighting factor of each combination is determined by the sum of the pools it contains. For example, you set the following criticality ratings:

• Usage: High
• Frequency: Low
• Growth Factor: Medium

The default weighting factor of the combination USAGE, FREQUENCY, GROWTH FACTOR is equivalent to $3 + 1 + 2$. Therefore, the default weighting factor for this combination is 6.

CCLM - How to setup 62
The default value for each intersection can be adjusted to a custom factor if needed.

⚠️ **Attention:** Each object is only included in the combination for which it contains the most relevant data for the highest number of data pools. For example, if an object has relevant data for the combination USAGE, FREQUENCY, GROWTH FACTOR, the object is not included in the combination USAGE, GROWTH FACTOR, even though it is relevant for both of those intersections.

### Schedule SOLAR project (step "Project Analysis")

This step only appears if one of the data pools "Table Utilization" or "Data Growth" was selected.

If DVM data extractors are running then DVM records the data growth and the table utilization. DVM does not record the reports or transactions, which are using the respective tables.

In order to the link reports and transactions (in fact the reports, which are assigned to the transaction) to the tables a project has to be defined in SOLAR01, where all the transactions and reports are assigned. One or more of these projects can be selected as a basis for the analysis.

⚠️ **Attention:** Only the reports and transactions assigned to the selected project(s) will be evaluated and checked for data growth and table utilization.

### Schedule criticality job (step “Schedule Job”)

For each system there exists a specific criticality job to calculate the criticality for the different objects. Each job must be scheduled separately in the set-up. This job is selecting the critical objects out of the different data pools. It is registering for each object, for which data pool this object is marked as critical, to be able to identify the intersection in which this object is included. Base on the weightage for each intersection a criticality can now be calculated.

Remember: an object is rated as critical for a data pool if it is part of the upper or lower threshold defined before.

The calculation of the criticality value for the objects is based on the defined weightage and the intersections an object is included. This can be a number from 0 to 100.

For CCLM the highest number will be calculated and then the number range will be divided into 5 equal ranges (range 1 is the lowest range starting from 0; range 5 is the highest range ending at highest number). It is now possible to assign every object to one of these ranges. The number of the range is now the indicator for the criticality in CCLM. This means, that for CCLM the following criticality values are available:

- SAPCriticality not applicable; for all objects which cannot be evaluated (all objects which are not of type PROG or TRAN)
- SAPCriticality = 0; object is not part of any intersection of data pools or is not critical in any data pool -> not critical
- SAPCriticality = 1; objects is part of at least one intersection and calculated number is in range 1 -> low criticality
• SAPCriticality = 2, SAPCriticality = 3, SAPCriticality = 4, objects is part of at least one intersection and calculated number is in respective range -> the higher the number the higher the criticality
• SAPCriticality = 5, objects is part of at least one intersection and calculated number is in range 5-> very criticality

In order to make the city model more readable the SAPCriticality=0 and SAPCriticality=1 will be consolidated in Criticality = 1.

The collection of the criticality into CCLM attribute SAPCriticality is done by the criticality collector scheduled in step 7.2

Although there is a calculation for each object in each system the overall criticality, which will be recorded in CCLM for each object is system independent. In CCLM the highest criticality level in a system for an object will be recorded.

⚠️ **Attention:** Even if a specific system is selected in the city model of the CCLM workcenter the highest criticality level of an object will be assigned to the object (system independent).

**Troubleshooting / Q&A**

**Q: What is the purpose of duplicates? Is it a problem if I have some of them?**
**A:** Duplicates indicate you have custom code objects with the same technical name across your whole development landscape and which was not created in your lead (development) system. e.g. a customer program called Z_HELLO_WORLD exists in your lead (development) system for an ERP system and at the same time in a development system for a CRM system. This means that these two objects with the same name have different source systems and both programs have different purposes. This is an indicator for an optimization of governance of custom code. Maybe your development system for CRM has to be flagged as lead system as well or you have to think about how to manage development for CRM independently or depending on the development for ERP.

**Q: How can I maintain organizational units?**
**A:** Organizational Units are needed when creating owners and contracts. They can be maintained using transaction SM30 for table AGS_ORGUNIT_CUST.

**Q: Can I filter the object view by a system?**
**A:** With the object collector run, a standard attribute field REF_2_SYSTEM will be maintained automatically. This will give you the needed detail level of attributes.

**Q: We have a registered customer namespace, but objects under that namespace are not collected by CCLM?**
**A:** Only objects in namespaces registered as producer (P) are collected. Table TRNSPACET contains the customer namespaces with their respective roles. In the development system in table TRNSPACET you should see the value ROLE = ‘P’ for your customer namespace(s). In the production system this flag may not be maintained, because it is not desired to change or create objects in the production system.

Therefore you need to let the system know that objects under this namespace need to be collected, despite of not having the producer role. For this purpose, you can use the function module /SDF/CMO_FILL_NSPACE to register this exception in the production system.

Example: Customer Namespace "/ABC/".
Call function module /SDF/CMO_FILL_NSPACE and execute it (F8).
Enter "/ABC/" in the field IV_NAMESPACE and keep the field IV_NS_INFO with the proposed value "CUST". Execute (F8)

After this is set, reschedule the object collection for the system where you did the change. Objects in the registered namespaces will now be considered as customer objects and collected.

**Q: How to delete the whole library with all data?**
**A:** Use the report RAGS_CC_DELETE_LIBRARY to delete the library definition or the library instance data.

**Q: How to repair TADIR inconsistency?**
**A:** Use the activity TADIR inconsistency in CDMC or the Transaction STDR (Object Directory Consistency Check)

**Q: If starting transaction CCLM opens DVM workcenter instead of Custom Code library workcenter,**
**A:** Do the following:

- SSO credentials are required to start the CCLM.
- The NO_SAPSSO flag must be set to 'X' in the customizing table 'DSWPGLOBALCUSTOM'

**Q: Why does starting transaction CCLM display the message "Service cannot be reached" - "403 Forbidden"?**

**A:** The CCLM workcenter is an ABAP Web Dynpro based application. Certain HTTP services must be activated in transaction SICF before.

Please check first if in the Solution Manager Set-Up in "System Preparation" step 5.5 "Configure Gateway" has been executed successfully.

If the gateway has been configured and these services are still not active, the above mentioned error message is displayed, activate the ICF service in one of the following ways:

- Activate the following ICF services by starting transaction SICF
- Press key F8 to get an overview
• Navigate to Default host->SAP->BC->AGS_WORKCENTER-> right mouse click->'activate service' and
• Navigate to Default host->SAP->BC->WEBDYNPRO->SAP->AGS_CUSTOM_CODE-> right mouse click->'activate service'

To activate the City model the service /sap/bc/ui5_u15/sap/citymodel has to be activated in SICF.

**Q: City model does not show any data what do I have to do?**

**A:** Please check if the extractor for the „city model“ has already successfully collect data. If yes and the data will not be presented in the city model please implement note 2086697.

Alternatively you can manually execute the following steps:

1. Start customizing of Solution Manager

   Structure
   
   - SAP Solution Manager Implementation Guide
     - SAP Solution Manager
       - SAP Customizing Implementation Guide
         - Activate Business Functions
           - Gateway Service Enablement
             - Gateway
               - Odata Channel
                 - Configuration
                 - Administration
                   - General Settings
                     - Activate and Maintain Services
                       - Assign SAP System Aliases to OData Service
                       - Define Virus Scan Profiles
                       - Logging Settings
                       - Cache Settings
                       - Support Utilities

3. Select service "AGS_CC_ODATA_CITY" and remove system alias "LOCAL" and replace it with "LOCAL_SM"

![Activate and Maintain Services]

![System Aliases]
4. Add alias "LOCAL_SM" with F4 help

5. Save changes

Q: How to identify new or deleted custom code objects?
A: The automatic collector job will read the current available custom code objects. As soon as it was found the name of the object will be registered in the library and in addition a link to this managed system will be created under the attribute REF_2_SYSTEM. So every registered custom code object will stay forever in the library and the existence in a managed system can be evaluated with the number of REF_2_SYSTEM attributes for this specific object. If the object is no longer found in the managed system the attribute REF_2_System will receive an end date. As soon as the object couldn't be found in any managed system anymore after a defined period of time (Customizing: CHECK_REF_DATE, Standard = 10 days) it will be marked as deleted in CCLM with setting attribute CustomerDeletion Flag to 'X' an the SAPLifecycleStatus to '99'.

💡 As soon as the CustomCodeObjects will be found again with the same name in a lead system or if the object was recreated again, the attribute 'DeletionFlag' will be set to initial. So all objects in library with an attribute assignment 'DeletionFlag' but with initial value are restored and new created custom code objects with the already existing name.

Q: Can I also add my own main object types similar to system, contracts,...?
A: From a logical point of view this is possible. But from a UI integration perspective this will not be without a modification and enhanced Webdynpro maintenance views.

Q: Can I also add manually custom code objects to the CCLM?
A: Technically this is possible by using the correct sequence of CCLM APIs and methods. But we do not recommend doing it because it will not fit to the main purpose of CCLM to have a full automated custom code objects collection without any basic manual effort. The datamodel is able to handle it.

Q: Can I have my one attribute with search help and value check?
A: Yes this is possible. Please create your own library attribute and ensure to set the flags VALUE check and add also a customer specific search help. Copy an existing search help like AGS_CC_SHLP_CRITICALITY, replace the value for the IDENTIFIER in the search help definition and add your wished values to the table AGS_CCD_SHLP.

Q: If I search objects via attributes, not all objects are displayed in fact the result is limited to 500 objects?
A: This configurable with table AGS_CC_CUSTOM field FILTER_LIMIT. Standard is 500. This field can be changed to any other value. If it will be empty there is no restriction.
Q: Why can’t I work with an object in the object list of a decommissioning project?
A: If an object is greyed out in the object list and it is not possible to work with this object for decommissioning in this project, then it is already part of another decommissioning project. Select the objects and press the "details" button to check the Decommissioning project to which this object is assigned.

Q: In SolMan_Setup for Custom Code Management in step 6.2 I got the error message "inactive UPL extractor" while trying to activate the UPL extractor. How should I proceed?
A: The following workaround will solve the issue. Above the table with the extractor in step 6.2 there is a link "Extractor Framework Admin UI" Click on this link. You will be directed to the List of all extractors. Please filter in column "Extractor" for "*UPL*" and in column "System" with the system ID. You will get one extractor with NOT "(Multicube Architecture)" in the description. Change into edit mode (button on the top) and delete this extractor.

After that try again to activate the UPL extractor in step 6.2.

Q: With SP11 the direct RFC collection was replaced by BW cube selections. Where does the data extraction from managed system happen?
A: This happens now based on SAP standard extractor framework. You can check each extractor in the extractor framework UI, which can be accessed in the set-up step 6.2.

Q: After upgrade to SP11 the Report SM_CCL:OBJECTS_SID_00000000.. dumps with the error that it is not possible to create an attribute or object?
A: With the upgrade of SP11 the application switch from using an internal number range to the standard ABAP number range. Investigating the issue: Starting Transaction SNRO, enter the number range "AGS_CC_ATR" or "AGS_CC_OBJ" and push the button "Number Range" and with pressing the button with the "Overview" icon. Now the number range is visible. If the number range is empty the Solman_Setup has not been executed and the CCLM has not been started. Short solution is to start transaction CCLM but the customer should execute the solman_setup to configure CCLM correct.

Q: When filtering for objects in the object view I don’t get results for methods?
A: Methods contain a lot of blanks in their object name. When displaying these objects in the object view the blanks where cut by the system to just one blank. Filtering the objects now with just this one blank in the object name doesn’t lead to a result. Please, when filtering exchange the blank with a "*" to get the right results.

Q: When setting-up CCLM the object collector returns "no objects found" when running it directly after the set-up (step 7.2)?
A: From SP11 onwards the collection of objects will be done in two separate steps. First step is to extract the data from the managed system and load it into BW cubes. Second step is to collect the objects out from the BW cubes and transfer the information into CCLM. Please make sure that the extractors defined in step 6.2 already extracted data BEFORE the collector of step 7.2 start.

Q: Does the CCLM read Code inspector results or ATC results from the managed system?
A: To get the quality data into CCLM an ATC master run must be scheduled on the managed system. This means the new and strategic ATC tool must be implemented in the respective system for all new CCLM installations from SP11 onwards. For existing CCLM installation Code-Inspector results can be considered if the collector configuration for the quality collector in the Solman Setup for Custom Code Management step 7.1 is
set to "RFC to Managed System". It is recommended to implement ATC and use the "local" mode in the set-up.

Q: I applied for ATC exemptions, will the CCLM exclude ATC results/findings for which exemptions have been applied?
A: The requested exemptions have to be approved by the Quality Manager. If it has been approved, then the results of the next ATC run will exclude the findings of the objects containing approved exemptions. The new ATC results (excluding the exempted findings) are then pulled into CCLM when the quality job gets executed.

If the user only applies for an exemption or if the exemption is rejected by the quality manager, then the results are not excluded from CCLM.

Click on ATC and ATC Exemptions to know more about them.

Q: Collector job was canceled with error message "Child collector is still active for <system ID> <Inst No>. What happened?
A: If a collector Job was cancelled with this error message, then another collector job was already running in the system. It is not possible that two collector jobs for CCLM are running at the same time. The cancelled job just reschedules itself, so please wait for the next running. If the data is needed then you can also activate the scheduled job. Please make sure that the collector jobs are running not in parallel.

Q: Collector job did not find any new data, what do I have to do?
A: Either there is no new data then there is nothing to do. If there should be new data, then please make sure that the extractor is running first to get the new data into Solution Manager BW cube.

Q: I don't get any ATC data in CCLM, what's wrong?
A: In order to get ATC data, ATC must be implemented in the managed system(s) and a master run has to be scheduled. Before data gets into CCLM the master run has to be executed successfully and AFTER this master run the extractor has to extract the master run results into Solution Manager BW. If the data is in BW then the collector is able to collect the data into CCLM.

Q: Collector job for last usage canceled with query execution error, what do I have to do?
A: Please implement and execute note 2034830.

Q: The object extractor shows read light for the data loader of the objects. Message is "illegal input ....", what do I have to do?
A: With SP11 the objects will be loaded into a Solution Manager BW cube. In some cases the description of the objects contains characters, which are not allowed for BW. The illegal character is shown in the message itself. You can add this character to the permitted characters with transaction RSKC. Please implement note 2110199.
Appendix

Special customizing of CCLM

The followings section contains parameters to influence the Custom Code Library.

⚠️ **Attention:** It is strictly recommended not to play with any parameters without knowing the impact.

Default settings of the CCLM extractor jobs in the table `AGS_CC_EXTRACTOR` defined string. This supports the feature to have several extractors for customer attributes.

Define lead system(s) (Development System) and do the decision for local object (Production System).

Default settings of CCLM basic customizing in table: `AGS_CC_CUSTOM`

<table>
<thead>
<tr>
<th>CC_KEY</th>
<th>CC_CONTENT (default value)</th>
<th>Do not modify</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADHOC_REPORTING</td>
<td>X</td>
<td></td>
<td>Activate the ABAP reporting link</td>
</tr>
<tr>
<td>BI_ANALYSIS</td>
<td>X</td>
<td></td>
<td>Activates the &quot;Dashboard&quot; View</td>
</tr>
<tr>
<td>BI_REPORTS</td>
<td></td>
<td></td>
<td>Activates the &quot;Reports&quot; View (obsolete with SP14)</td>
</tr>
<tr>
<td>BI_CONTRACT</td>
<td>X</td>
<td></td>
<td>Activates &quot;Contracts&quot;</td>
</tr>
<tr>
<td>BI_OWNER</td>
<td>X</td>
<td></td>
<td>Activates &quot;Owner&quot;</td>
</tr>
<tr>
<td>CHECK_REF_DATE</td>
<td>10</td>
<td></td>
<td>Retention time period in days for setting end date of the ref_2_system attribute (depends on CHECK_REFERENCE) and setting deletion flag of objects</td>
</tr>
<tr>
<td>FILTER_LIMIT</td>
<td>500</td>
<td></td>
<td>Limit the results of an attribute filtering</td>
</tr>
<tr>
<td>LOC_ATTRIBUTS</td>
<td>X</td>
<td></td>
<td>LOCAL or REMOTE if collector collects attributes from BW (local) or from managed system ()</td>
</tr>
<tr>
<td>LOC_CRIT</td>
<td>X</td>
<td></td>
<td>LOCAL or REMOTE if collector collects criticality</td>
</tr>
<tr>
<td><strong>CC_KEY</strong></td>
<td><strong>CC_CONTENT (default value)</strong></td>
<td><strong>Do not modify</strong></td>
<td><strong>Explanation</strong></td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------</td>
<td>-------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>LOC_LASTUSAGE</td>
<td></td>
<td>X</td>
<td>LOCAL or REMOTE if collector collects last usage from BW (local) or from managed system ()</td>
</tr>
<tr>
<td>LOC_OBJECTS</td>
<td></td>
<td>X</td>
<td>LOCAL or REMOTE if collector collects objects from BW (local) or from managed system ()</td>
</tr>
<tr>
<td>LOC_SCI</td>
<td></td>
<td>X</td>
<td>LOCAL or REMOTE if collector collects quality data from BW (local) or from managed system ()</td>
</tr>
<tr>
<td>LOC_SIMILARITY</td>
<td></td>
<td>X</td>
<td>LOCAL or REMOTE if collector collects similarity data from BW (local) or from managed system ()</td>
</tr>
<tr>
<td>Library_ID</td>
<td>Custom_Code_Library</td>
<td>X</td>
<td>&lt;&lt; database primary necessary for the active instance</td>
</tr>
<tr>
<td>Library_Key</td>
<td>CCL</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Mod_Range</td>
<td>365</td>
<td></td>
<td>The time range to fetch old inactive modifications : get only modifications of last 365 days. (only valid for the first run of object collector to fetch modifications) see note 2143772</td>
</tr>
<tr>
<td>NO_RETIREMENT_1</td>
<td>TABL</td>
<td>X</td>
<td>Decommissioning cockpit will not support objects of type TABL</td>
</tr>
<tr>
<td>NO_RETIREMENT_2</td>
<td>WDYA</td>
<td>X</td>
<td>Decommissioning cockpit will not support objects of type WDYA</td>
</tr>
<tr>
<td>NO_RETIREMENT_3</td>
<td>WDYN</td>
<td>X</td>
<td>Decommissioning cockpit will not support objects of type WDYN</td>
</tr>
<tr>
<td>TABS</td>
<td></td>
<td></td>
<td>Will activate the tabbed object view</td>
</tr>
</tbody>
</table>