representative if you require access to the modified Apache Software Foundation source files.

Licensees may duplicate the software product user documentation contained on a CD-ROM, but only to the extent necessary to support the users authorized access to the software under the license agreement. Any reproduction of the documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

U.S. GOVERNMENT RESTRICTED RIGHTS. It is acknowledged that the Software and the Documentation were developed at private expense, that no part is in the public domain, and that the Software and Documentation are Commercial Computer Software provided with RESTRICTED RIGHTS under Federal Acquisition Regulations and agency supplements to them. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 et. seq. or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at FAR 52.227-19, as applicable. Contractor is Micro Focus (IP) Ltd, 9420 Key West Avenue, Rockville, Maryland 20850. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.
Table of Contents

1. ABOUT THIS DOCUMENT .................................................................................................................. 7

2. PREPARATION ................................................................................................................................. 9
   2.1. PREREQUISITES ......................................................................................................................... 9
   2.2. UNDERSTANDING ANALYSIS PROCESS PHASES ...................................................................... 9

3. ENTERPRISE VIEW PROCESS OVERVIEW .................................................................................... 11

4. PROVIDING SOURCES TO THE ENTERPRISE VIEW SERVER MACHINE ........................................ 15
   4.1. INTRODUCTION .......................................................................................................................... 15
   4.2. ENTERPRISE VIEW EXTENSIONS FOR THE IMS BATCH ENVIRONMENT .................................. 16
   4.3. ENTERPRISE VIEW EXTENSIONS FOR THE IMS ONLINE ENVIRONMENT ............................... 17
   4.4. ENTERPRISE VIEW EXTENSIONS FOR THE CICS ENVIRONMENT ........................................... 17
   4.5. MISSING SOURCES .................................................................................................................. 18
       4.5.1. Missing Programs or Transactions ...................................................................................... 18
       4.5.2. Missing Copy/Include Books ............................................................................................... 18
       4.5.3. Missing JCLs and JCL Catalogued Procedures .................................................................... 18
       4.5.4. Missing IMS Stage One File or CICS Resource Definition File ........................................... 19
       4.5.5. Missing Database Description File ......................................................................................... 19

5. DEFINING AN ORGANIZATION VIEW TREE .................................................................................. 21

6. STARTING HOST JOBS ................................................................................................................... 23

7. STARTING THE INVENTORY PROBE JOB ................................................................................... 27
   7.1. INTRODUCTION .......................................................................................................................... 27
   7.2. HOW TO RUN THE INVENTORY PROBE JOB ........................................................................... 27
   7.3. UNDERSTANDING INVENTORY PROBE JOB RESULTS ............................................................ 29
   7.4. WHEN TO RERUN INVENTORY PROBE JOB .............................................................................. 31
       7.4.1. Rerun after Adding New Components ............................................................................... 31
       7.4.2. Rerun after Adding New Extensions .................................................................................. 31

8. STARTING INVENTORY JOB .......................................................................................................... 33
   8.1. INTRODUCTION .......................................................................................................................... 33
   8.2. HOW TO RUN INVENTORY JOB .................................................................................................. 33
   8.3. INVENTORY JOB RESULTS ....................................................................................................... 35
   8.4. WHEN TO RERUN INVENTORY JOB ........................................................................................... 37

9. STARTING ANALYSIS JOB ............................................................................................................. 39
   9.1. INTRODUCTION .......................................................................................................................... 39
   9.2. HOW TO RUN THE ANALYSIS JOB .......................................................................................... 40
   9.3. ANALYSIS JOB RESULTS .......................................................................................................... 42
       9.3.1. Unresolved Objects ............................................................................................................. 44
       9.3.2. Analysis status Done (with warnings) ................................................................................. 47
       9.3.3. Analysis status Done (with errors) ....................................................................................... 47
       9.3.4. Analysis status Done (timeout) ............................................................................................ 48
       9.3.5. Analysis status Done (not parsed) ....................................................................................... 48
       9.3.6. Sources with status To Do (Extended) after analysis job .................................................... 48
   9.4. WHEN TO RERUN THE ANALYSIS JOB .................................................................................... 49

10. STARTING LIFE CYCLE JOB ......................................................................................................... 51
10.1. INTRODUCTION............................................................................................................... 51
10.2. HOW TO RUN LIFE CYCLE JOB.................................................................................... 51
10.3. UNDERSTANDING LIFE CYCLE STATUS RESULTS...................................................... 53
11. STARTING THE UNKNOWN SYNTAX JOB....................................................................... 55
  11.1. INTRODUCTION ............................................................................................................ 55
  11.2. HOW TO RUN THE UNKNOWN SYNTAX JOB ............................................................... 55
  11.3. UNDERSTANDING UNKNOWN SYNTAX RESULTS....................................................... 55
12. STARTING THE INVENTORY DELETION JOB................................................................. 57
  12.1. INTRODUCTION ............................................................................................................ 57
  12.2. HOW TO RUN THE INVENTORY DELETION JOB ......................................................... 57
13. STARTING THE HTML EXPORT JOB ............................................................................. 59
14. STARTING THE END DAEMON JOB ............................................................................. 61
15. ANALYZING DL/I SOURCES.......................................................................................... 63
  15.1. IMS DB DL/I GENERAL CONSIDERATIONS................................................................. 63
  15.2. SOURCES CONTAINING DL/I DATABASE DEFINITIONS ............................................. 63
    15.2.1. Introduction ............................................................................................................... 63
    15.2.2. DL/I Descriptor Language ....................................................................................... 64
    15.2.3. VSAM Delete/Define File Statements ....................................................................... 66
    15.2.4. JCL Physical Files ...................................................................................................... 67
16. DL/I PROGRAM LANGUAGE INTERFACE.................................................................... 69
  16.1. INTRODUCTIONS .......................................................................................................... 69
  16.2. EXEC DLI COMMAND INTERFACE .............................................................................. 69
  16.3. CALL DLI LANGUAGE INTERFACE .............................................................................. 69
17. ENTERPRISE VIEW ANALYSIS OF DLI COMMANDS .................................................. 71
  17.1. SOURCES WITH BASIC ANALYSIS ............................................................................. 71
  17.2. SOURCES WITH EXTENDED ANALYSIS ..................................................................... 71
1. About this Document

This guide explains the basic concepts of Enterprise View analysis processing of applications working with IMS DB DL/I databases on mainframe platform (z/OS, OS/390, MVS). IMS DB DL/I is the database portion of the IMS (Information Management System) product and operates in conjunction with the transaction management portion of IMS (IMS TM or DC) and CICS (Customer Information Control System). It can also operate in a standalone-batch environment.

This guide is intended for Enterprise View users who develop applications working with IMS DB DL/I databases from IMS or CICS environments. The applications to be analyzed have to be written on program languages that the Enterprise View product supports on mainframe platforms: Assembler, C/C++, COBOL, Cobol Delta Macro and PL/I. This guide describes how to define and analyze such applications using the Enterprise View product and keeping Enterprise View requirements for analysis processing.

You can read this guide from beginning to end to get a full understanding of the topics. Alternatively, advanced users can read the chapters individually because they are self-contained.
2. Preparation

2.1. Prerequisites

In order to configure and analyze applications, the following software must be installed:

1. Enterprise View Host Part (Server Modules) on the Server machine
2. Enterprise View Client Part (Client Solutions) on the Client machine
3. Oracle Database on the Server machine
4. Enterprise View MVS Agent (optional)

Enter a User Identifier and Password to ensure:

1. access to the Enterprise View Host Server machine.
2. the Enterprise View Client Solutions will start on the client PC.
3. access to the Oracle Database Schema on the Server machine.

For more information about installation, administration and configuration of these products refer to:

- Enterprise View Installation Guide
- Enterprise View Administration Guide

2.2. Understanding Analysis Process Phases

A process overview of the analysis phases is given in the next chapter. It can be used to better understand the process of an analysis of the applications working with IMS DL/I database. The overview shows the sequence that you must follow to successfully execute the entire process. Depending on the messages and the different cases of an analysis, some of the steps can be repeated.

The analysis process has two parts:

1. Primary analysis process of the application.
2. Post-analysis (life cycle) process of the analyzed application. During this phase the applications are checked for modifications, for example, new sources, deleted ones, or changed program code. The Enterprise View Repository and Database tables are updated with the new modifications.

The Inventory and Analysis Jobs for a given defined organization view must be executed until successful completion of the primary analysis process. The client can repeat the execution of the Inventory Job only after executing the Inventory Deletion Job. The Life Cycle Job is executed when modifications have been made to the applications after successful completion of the primary analysis process.
3. Enterprise View Process Overview

Part I. Primary analysis process overview

- Transfer sources to Enterprise View Server machine
- Identify source code and check file extensions
- Define Technical View
- Define Custom View (Optional)
- Run Inventory Probe Job
- Check if all necessary components are in the package
- Check for unsupported files and wrong extensions
- Work with file extensions
- Run Inventory Deletion Job with Delete Structure
- Add the components
Part II. Post-analysis (Life Cycle) process overview

1. Select the Organization View (Already successfully analyzed)

2. Run Life Cycle Job

3. Check the results for:
   - 1. Unresolved Objects
   - 2. Error messages

4. Add missing sources

5. Check for Proposed Dynamic Links

6. Correct the errors

7. Confirm Dynamic Links

8. Client Modules:
   - Enterprise View Tech
   - Enterprise View Builder

Transfer modified application sources to the Enterprise View Server machine in the corresponding components of the Organization View.
4. Providing Sources to the Enterprise View Server Machine

4.1. Introduction

Applications using the IMS DL/I database can be created and tested directly in the mainframe environment (z/OS, OS/390, MVS). Using Micro Focus Mainframe Express, you can create applications using the IMS DL/I database in a Windows environment (Windows 95/98, Windows 2000, and Windows NT 4.0) and test them on the IMS/ESA platform.

If the sources are not on the Enterprise View server machine they must be transferred in order to be analyzed. We recommend you use FTP (File Transfer Protocol). You can also use the Enterprise View MVS Agent Module when applications are located on a mainframe platform. If you do not have Enterprise View MVS Agent installed, you can use any other transfer method and protocol.

The transferred sources can be located in one or more folders. We strongly recommend that you download the sources using the same folder organization as they have on the source system. Thus the structure of the libraries will be better represented and will be closer to the developer’s view. Every folder must contain only one type of source code. For example, COBOL copy books must be in one folder, IMS/TP PL/I programs in another, JCL programs in another and so on. In addition, each type of source must be transferred with an extension recognized by the Enterprise View Knowledge Base. For ease of use, the module Enterprise View MVS Agent can be installed to allow quick configuration of the sources that will be transferred to the Enterprise View Server. This module allows partitioned selection and automatic extension assignment.
The client can delete existing extensions and create new extensions using the Work with File Extension menu on the Enterprise View Configurator. The source extensions must match the newly created extensions.

The highlighted extensions in the tables below are the Enterprise View extensions. They are grouped depending on the system and the environment in which the applications are performed. They can be performed in the following environments:

- **Batch environment:** [IMS Batch](#)
- **Online environment:** [IMS Online](#) or [CICS Online](#)

You must assign an extension or the sources will not be recognized.

**NOTE:** Some of the programs can have the common extensions: `.c` or `.C`, `.cbl` or `.CBL`, `.pli` or `.PLI`, and the other programs can have extensions that specify the environment where the programs run (batch or online), for example: `.cims` or `.CIMS`, `.cblims` or `.CBLIMS`, `.pliims` or `.PLIIMS`. The sources are handled in the same way by the Enterprise View parsers, but we recommend you use an extension linked to the environment as it is easier to display the data in the client modules (i.e. Enterprise View Tech and Enterprise View Builder) because the column Type will show different values related to the environment (i.e. IMS Batch or IMS Online). The tabsheet Overview will show the program distribution by type and it will be easier to extract the programs, applying the Filter Active function in the Type column.

### 4.2. Enterprise View Extensions for the IMS BATCH Environment

The extensions described in the following table are pre-defined extensions in the product. The user can use them for applications containing IMS programs and sources running in batch environment.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programs</td>
<td>IMS Assembler programs</td>
<td>.asmims, .ASMIMS</td>
</tr>
<tr>
<td></td>
<td>IMS C Programs</td>
<td>.c, .cims, .C, .CIMS</td>
</tr>
<tr>
<td></td>
<td>IMS C++ Programs</td>
<td>.cpp, .cppims, .CPP, .CPPIMS</td>
</tr>
<tr>
<td></td>
<td>IMS COBOL Programs</td>
<td>.cbl, .cblims, .CBL, .CBLIMS</td>
</tr>
<tr>
<td></td>
<td>IMS COBOL Delta Macro</td>
<td>.cdm, .cdmims, .CDM, .CDMIMS</td>
</tr>
<tr>
<td></td>
<td>IMS PL/I Programs</td>
<td>.pli, .pliims, .PLI, .PLIIMS</td>
</tr>
<tr>
<td>JCL</td>
<td>Job Control Language</td>
<td>.jcl, .JCL</td>
</tr>
<tr>
<td></td>
<td>Catalogued Procedure</td>
<td>.proc, .PROC</td>
</tr>
<tr>
<td></td>
<td>JCL Card</td>
<td>.card, .CARD</td>
</tr>
<tr>
<td>Copy/Include</td>
<td>Assembler Include Books</td>
<td>.asminc, .ASMINC</td>
</tr>
<tr>
<td></td>
<td>C Include Books</td>
<td>.h, .H</td>
</tr>
<tr>
<td></td>
<td>C++ Include Books</td>
<td>.hpp, .HPP</td>
</tr>
<tr>
<td></td>
<td>COBOL Copybooks</td>
<td>.cpy, .CPY</td>
</tr>
<tr>
<td></td>
<td>PL/I Include Books</td>
<td>.inc, .INC</td>
</tr>
<tr>
<td>Files</td>
<td>DL/I DBD</td>
<td>.dbd, .DBD</td>
</tr>
<tr>
<td></td>
<td>DL/I PSB</td>
<td>.dbpsb, .DBPSB</td>
</tr>
<tr>
<td></td>
<td>VSAM Delete Define</td>
<td>.ddef, .DDEF</td>
</tr>
</tbody>
</table>
**NOTE:** If any of these extensions is missing from the Enterprise View Knowledge Database, the client can create the same or other extensions using Environment Settings, Work with File Extensions menu on the Enterprise View Configuration Manager Module. This operation can be performed only by an administrator of the product.

### 4.3. Enterprise View Extensions for the IMS Online Environment

The extensions described in the following table are pre-defined extensions in the product. The user can use them for applications containing IMS programs and sources running in online environment.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transactions</strong></td>
<td>IMS Transactions</td>
<td>.trn .TRN</td>
</tr>
<tr>
<td><strong>Programs</strong></td>
<td>IMS/TP Assembler programs</td>
<td>.asmims .ASMIMS</td>
</tr>
<tr>
<td></td>
<td>IMS/TP C Programs</td>
<td>.cims .CIMS</td>
</tr>
<tr>
<td></td>
<td>IMS/TP C++ Programs</td>
<td>.cppims .CPPIMS</td>
</tr>
<tr>
<td></td>
<td>IMS/TP COBOL Programs</td>
<td>.cblims .CBLIMS</td>
</tr>
<tr>
<td></td>
<td>IMS/TP Cobol Delta Macro Programs</td>
<td>.cdmims .CDMIMS</td>
</tr>
<tr>
<td></td>
<td>IMS/TP PL/I program</td>
<td>.pliims .PLIIMS</td>
</tr>
<tr>
<td><strong>IMS/Resource Definitions</strong></td>
<td>IMS Stage One</td>
<td>.ims .IMS</td>
</tr>
<tr>
<td><strong>Include Books</strong></td>
<td>Assembler Include Books</td>
<td>.asminc .ASMINC</td>
</tr>
<tr>
<td></td>
<td>C Include Books</td>
<td>.h .H</td>
</tr>
<tr>
<td></td>
<td>C++ Include Books</td>
<td>.hpp .HPP</td>
</tr>
<tr>
<td></td>
<td>Cobol Copy Books</td>
<td>.cpy .CPY</td>
</tr>
<tr>
<td></td>
<td>PL/I Include</td>
<td>.inc .INC</td>
</tr>
<tr>
<td><strong>Map Definitions</strong></td>
<td>MFS Definitions</td>
<td>.mfs .MFS</td>
</tr>
<tr>
<td></td>
<td>MFS Map Definitions</td>
<td>.mapmfs .MAPMFS</td>
</tr>
</tbody>
</table>

**NOTE:** If any of these extensions is missing from the Enterprise View Knowledge Database, the client can create the same or other extensions using Environment Settings, Work with File Extensions menu on the Enterprise View Configuration Manager Module. This operation can be performed only by an administrator of the product.

### 4.4. Enterprise View Extensions for the CICS Environment

The extensions described in the following table are pre-defined extensions in the product. The user can use them for applications containing CICS programs.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programs</strong></td>
<td>CICS Assembler programs</td>
<td>.asmcic .ASMCIC</td>
</tr>
<tr>
<td></td>
<td>CICS C Programs</td>
<td>.ccic .CCIC</td>
</tr>
<tr>
<td></td>
<td>CICS C++ Programs</td>
<td>.cppcic .CPPCIC</td>
</tr>
<tr>
<td></td>
<td>CICS COBOL Programs</td>
<td>.cblcic .CBLCIC</td>
</tr>
</tbody>
</table>

DLI Process Guide
| CICS COBOL Delta Macro Programs | .cdmcic, .CDMCIC |
| CICS PL/I program               | .plicic, .PLICIC |
| CICS/Resource Definitions      | CICS Resource Definition File .csd, .CSD |
| Include Books                  | Assembler Include Books .asminc, .ASMINC |
|                                | C Include Books .h, .H |
|                                | C++ Include Books .hpp, .HPP |
|                                | COBOL Copy Books .cpy, .CPY |
|                                | PL/I Include .inc, .INC |
| Map Definitions                | Map Definitions .map, .MAP |
|                                | BMS Map Definitions .mapbms, .MAPBMS |

**NOTE:** If any of these extensions is missing from the Enterprise View Knowledge Database, the client can create the same or other extensions using Environment Settings, Work with File Extensions menu on the Enterprise View Configuration Manager module. This operation can be performed only by an administrator of the product.

### 4.5. Missing Sources

If sources are not provided, they cannot be analyzed. When source is missing but is referenced by any other source, an unresolved object is created. Many other issues can also be generated. The most important issues are listed below.

#### 4.5.1. Missing Programs or Transactions

*Issues:*
- An unresolved object of specific type will be generated.
- Any recognized call to missing objects will link to the unresolved object.
- Technical Function Point will be corrupted.
- Impact Analysis will not be possible.

#### 4.5.2. Missing Copy/Include Books

*Issues:*
- If referenced, an unresolved object of specific type will be generated.
- Metrics without copy/include books will be corrupted.

#### 4.5.3. Missing JCLs and JCL Catalogued Procedures

*Issues:*
- Since the JCLs are starting points, the chain between the JCL and the objects referenced to and from missing JCL will be interrupted.
- The programs called by the JCLs or procedures will not be referenced.
- The files used will not be referenced.
- The relationships among files and programs declared in the missing procedures will be lost.
- There will be a missing link between a physical file and the program using it.
- There will be missing information about the link between the files and the data areas describing the file.
4.5.4. Missing IMS Stage One File or CICS Resource Definition File

Issues:
- The transactions and programs being declared will be lost.
- The relationships between transactions and programs being declared will be lost.

4.5.5. Missing Database Description File

Issues:
- The database list will be lost.
- The segment list will be lost.
- The segment’s hierarchy will be lost.
5. Defining an Organization View Tree

The Organization View is an Enterprise View structure. It gives a better representation of the structure of the sources. The client can organize his applications in Technical or Custom Organization Views. The Technical organization view is used for:

- Starting the jobs: Inventory, Analysis, Life Cycle, Delete Inventory etc.
- Better presenting and understanding the results and the charts after the analysis job

You can also create a Custom organization view. This enables you to see the analysis results only, from a different point of view, reordering and connecting the components from one or more technical organization views in one custom view. You can combine components by criteria, different from developer's point of view and closed to his business environment.

For further details about the goals and how to define an organization view, see the “Organization View Process Guide”.

You should bear the following in mind when defining an organization view:

- You must select system OS/390 when defining a Technical organization view for sources running in IMS and CICS environments on a mainframe platform.
- A Custom organization view can be created at any time during the analysis process.
6. Starting Host Jobs

There are two ways to submit the host jobs for execution: using the ‘Work with Jobs’ icon or ‘Organization View Tree’. They are described individually in the chapters for every host job. In the newer version of the Enterprise View product, a Job Creation Wizard creates the starting parameters of the host jobs. This Job Creation Wizard can also be started from the ‘Work with Jobs’ icon or ‘Organization View Tree’. The sequence of the Job Creation Wizard menus is presented below:

1. The Job Wizard Welcome menu for submitting a host job. Press Next to continue or Cancel to exit the wizard.

2. The Job Wizard Selection menu that contains all the different types of jobs. You can select the name of the host job by choosing from the drop-down list. You can also set the schedule date and time. You can click Next to continue or Back to review your previous choices. If you choose the Cancel button, the Job Wizard will terminate. The host jobs to be examined are:
   - Inventory probe Job
   - Inventory Job
   - Analysis Job
   - Life Cycle Job
   - Inventory Deletion Job
   - HTML Export Job
   - End Daemon Job
3. The Job Wizard menu for selecting parameters: packages, organization views, operational modes and current period are available from the drop-down lists. ‘Update Trend’ is used if you are interested in the previous status and the movement of the metrics of the objects. You can click Next to continue or Back to review your previous choices. If you choose the Cancel button, the Job Wizard will terminate.

4. The Job Wizard menu for selecting kbXDaemon, to will execute the submitted job, and the parameters of the submitted job. There are two Daemon jobs: MAIN and SECONDARY. You can click Generate to continue or Back to review your previous choices. If you choose the Cancel button, the Job Wizard will terminate.
4. The Job Wizard Complete menu. Press Finish to exit the Wizard.

After clicking the Finish button, the job will be submitted for execution when the Job Wizard is started from the organization view tree. Otherwise, the ‘Work with Jobs’ menu will appear. Depending on the Daemon name selected in the Wizard menu, you have to start the appropriate kbXDaemon job and press the OK or Apply button to submit the job.
7. Starting the Inventory Probe Job

7.1. Introduction

The Inventory Probe Job verifies if all sources provided by means of the Technical Organization View Definition have extensions recognized by the Enterprise View Knowledge Base. The Inventory Probe Job can be run only at the highest level of the organization view - the package.

As described in the chapter “Providing Sources to the Enterprise View Server Machine”, the product requires the sources which are to be processed to be placed in different folders, depending on the kind of sources. They must have the proper extension for the type of source.

This job answers the following questions:

- How many sources are contained in a package?
- How many sources of each type are contained in a package?
- What types of sources are contained in the package?
- Are there extensions (i.e. type of sources) which are not supported?

The Inventory Probe Job produces a report that contains the count of sources in the package by extension, divided by catalogued and uncatalogued ones.

**NOTE:** Although you do not have to do this, we strongly recommend you do because it checks at an early stage if all the source types necessary for the process (Inventory and Analysis) have been included in the package.

This job can be repeated if necessary. For more information see the chapter [When to Rerun Inventory Probe Job](#).

7.2. How to Run the Inventory Probe Job

There are two ways to run the Inventory Probe job. In both cases, Job Wizard menus will start if you work with the newer versions of the Enterprise View Navigator.

1. **Using the ‘Work with Jobs’ menu:**

   - Click on the Work with Jobs icon 🗂️ and select the New button. If the Job Creation Wizard menu appears, go to the chapter [Starting Host Jobs](#). Select Inventory Probe from the pop-up window and press the OK button. The Work with Jobs menu can be selected using the Enterprise View Configuration Manager, Enterprise View Builder and Enterprise View Tech solutions.
The window with fields which are not filled in will appear on the screen.

Fill in the fields as described in the chapter “Work with Jobs” in the “Common Features” reference guide. The Inventory Probe Job with the filled-in fields is ready for execution.

2. Using the Organization View tree:

- Select a package in the organization view tree.
- Right-click the selected package, choose the Job Submission item and select Submit Inventory Probe Job from the pop-up menu as shown in the following picture. If the Job Creation Wizard menu appears, go to chapter 6, Starting Host Jobs.
In both cases the Work with Jobs window will appear with the information about the selected package.

- Start the KbXDaemon session on the server machine.
- Press the OK button from the Work with Jobs window to start the job.

After execution of the job, select Work with Jobs again. The Show Result button appears. Press the Refresh button, and then the Show Result button. The following window with the results from the execution will be displayed:

![Image of results window]

### 7.3. Understanding Inventory Probe Job Results

The results from the execution of the Inventory Probe job are written to a .txt file on the server machine. The sources (or their extensions) are classified into two main groups:

- **Catalogued**

  All objects in this category will be inventoried. Depending on the Analysis and Save mode, different action will be taken during the analysis phase.

  a) **Extended Analysis** - Sources corresponding to the extensions described in this group are fully analyzed by parsers with extended analysis.

     1 - Object Reference Analysis for all objects and Detailed Analysis only for the supported language is performed.
     2 - Sizing Measure
     3 - Quality Measure
     4 - Objects Change Traffic Measure
     5 - LOC Change Traffic Measure
b) Basic Analysis - Sources corresponding to the extensions described in this group are analyzed by parsers with basic analysis.

1 - No Object Reference and Detailed Analysis
2 - Sizing Measure
3 - No Quality Measure
4 - Objects Change Traffic Measure
5 - LOC Change Traffic Measure

c) Disabled Analysis - Sources corresponding to the extensions described in this group are recognized in the Enterprise View Tables but only for Inventory purposes; the product does not perform any type of analysis after the Inventory phase.

1 - No Object Reference and Detailed Analysis
2 - No Sizing Measure
3 - No Quality Measure
4 - Objects Change Traffic measure (only New/Deleted)
5 - No LOC Change Traffic Measure

d) Unsaved Files - Sources corresponding to the extensions described in this group are recognized in the Enterprise View Tables but only for Inventory purposes; the product does not perform any type of analysis after the Inventory phase, and it is not possible to display their source code by means of the function Show Source or to trace source history because they have not been saved in the Enterprise View Repository.

1 - No Object Reference and Detailed Analysis
2 - No Sizing Measure
3 - No Quality Measure
4 - No Objects Change Traffic Measure
5 - No LOC Change Traffic Measure

Not Catalogued

All Objects belonging to this category will be ignored for the inventory.

a) Skipped Files - Sources corresponding to the extensions described in this group exist in the Enterprise View configuration tables, but objects with that extension are ignored. This group contains sources recognized in the Enterprise View Tables but not included in the Inventory or Repository.

b) Undefined Files - Sources corresponding to the extensions described in this group do not exist in the Enterprise View configuration tables, and are therefore not included in the inventory and are not analyzed.

This file gives a broad view of what is present in the package and helps to identify immediately if all the components necessary to process the user’s environment have been included in the package.
7.4. When to rerun Inventory Probe Job

You should rerun the Inventory Probe Job when there are no more incorrect or undefined extensions and missing sources.

7.4.1. Rerun after Adding New Components

If any sources have been omitted, it is important that you add them using ‘Define Technical View’ before proceeding with the other job. There are two ways of adding missing sources to the package and rerunning the Inventory Probe Job:

**I. Execute the Inventory Deletion Job**

1. Run **Inventory Deletion Job with Delete Structure** to delete the structure of the Technical Organization View for the defined application.
2. Create new folders for the missing sources in the main folder of the application on the server machine.
3. Transfer the missing sources to the new folders and check the file extensions. Some sources can also be transferred in the corresponding existing folders.
4. Define the Technical Organization View including the new folders as components.
5. Run Inventory Probe Job again and check the results.

**II. Don’t execute the Inventory Deletion Job (for advanced users)**

1. Create a new folder for the missing sources of the application on the server machine. If there are sources with different type file extensions, the folder can contain subfolders.
2. Transfer the missing sources to the corresponding new folders and check the file extensions.
3. Define the new Technical Organization View for these new folders only.
4. Attach the new defined package and its components at the right place in the existing organization view.
5. Run Inventory Probe Job again and check the results.

7.4.2. Rerun after Adding New Extensions

You must think carefully about the sources catalogued in “Undefined Files” because their extensions are unrecognized in the Enterprise View Knowledge Base. If these sources are necessary for the environment, their extensions can be added in the Enterprise View File Extension Table (for details refer to “Work with File Extensions” in the “Enterprise View Configurator” Reference Guide) and include them in the Enterprise View process, otherwise you can ignore or delete them from the source folders.

If the extensions shown in “Undefined Files” are due to having incorrectly assigned extensions to sources, you must correct them and rerun Inventory Probe Job.

**NOTE:** We recommend you rerun the Inventory Probe Job after adding new components or new extensions in the Work with File Extensions table to check that the Enterprise View product has properly recognized them.
8. Starting Inventory Job

8.1. Introduction

The Inventory Job loads information about all the sources in the folders provided by the users by creating organization views in the Enterprise View Tables and Repository. The Inventory Job can be run only at the highest level of the organization view - the package. There are three types of inventory scanners:

1. **Code Inventory**
   - The Code Inventory is a directory scanner that performs the following operations:
     - Scans the list of directories included in the Organization View
     - Generates a File Repository (a physical copy of all useful objects) in a separate directory:
       - Generates a ‘history’ version of all the useful objects (including the different output for different versions)
       - Generates a last version file and history of scans
     - Writes all the retrieved information to the Enterprise View Repository.

2. **Data Inventory**
   - Data Inventory is a database scanner that performs the following operations:
     - Scans the database attached to the Organization View and extracts the definitions of the tables, index, views, primary and foreign keys
     - Writes all the retrieved information to the Enterprise View Repository.

3. **Full Inventory**
   - The Full Inventory scanner performs Code and Data Inventory functions.

Performing an inventory of a package is a prerequisite to running the subsequent phases. The Inventory Job can be rerun as required. For more information, see the chapter *When to Rerun Inventory Job*.

8.2. How to run Inventory Job

There are two ways to run the Inventory Job. In both cases, Job Wizard menus will start if you work with the newer versions of the Enterprise View Navigator.

1. **Using ‘Work with Jobs’ menu:**
   - Click on the Work with Jobs icon and select the New button. If the Job Creation Wizard menu appears, go to the chapter *Starting Host Jobs*. Select ‘Inventory’ job type from the pop-up window and fill in the fields as described in the chapter “Work with Jobs” in the “Common Features” reference guide.
2. Using the Organization View tree:

- Select a package in the organization view tree.
- Right-click on the selected package, choose the Job Submission item and select Submit Inventory Probe Job from the pop-up menu as shown in the following picture. If the Job Creation Wizard menu appears, go to the chapter Starting Host Jobs.

In both cases, the ‘Work with Jobs’ window will appear with the information about the selected package.
Start a KbXDaemon session on the server machine.
Press the OK button on the ‘Work with Jobs’ menu to submit the job.

8.3. Inventory Job Results

The results of completing the Inventory Job can be viewed in the modules Enterprise View Builder, Enterprise View Tech, and Enterprise View Dynamic Inventory from different points of view. You can see:

- General and summary information about all loaded sources, by looking at the Overview and Summary tabsheets.
- Detailed information about all loaded sources, by looking at the specific tabsheet for each source class – Programs, Files, Copy/Includes, JCLs, Database Definitions and others.
- Program source code and object properties, by left-clicking on every row of the specific tabsheet for each source class and using the dropdown menu.

The next picture shows the ‘Summary’ tab sheet of the Enterprise View Builder solution.

Detailed information about all loaded objects in the All Objects tabsheet of the Enterprise View Builder is displayed in the next picture.
Information is written for every object in the Enterprise View Tables about:

- the qualification of the identification object:
  - name, alias and description
  - Enterprise View system, class, type, and language
- the metrics:
  - number of the objects
  - source lines of code (SLOC)
- the ‘unknown’ extensions
  - the objects with unknown extensions are marked as ‘skipped’. These objects are written in a file named skipped.txt in the JOB directory. The type of extensions and the number of ignored sources are recorded in this file. In the newer versions of the Enterprise View product the file skipped.txt is not created. Sources with ‘unknown extensions’ belong to the ‘uncatalogued’ category in the Inventory Probe job results. These sources exist inside the Enterprise View configuration tables, but they are ignored for the inventory.

The ‘Status’ field in the Enterprise View Builder and Enterprise View Tech client solutions shows the status of the every object after execution of the Inventory Job. This status is always New.
The ‘Analysis Status’ field in the Enterprise View Builder and Enterprise View Tech client solutions shows the results of the execution of the Inventory Job for every object in the selected Organization View.

When the Analysis Status field contains:

**To Do (Basic)** or **To Do (Extended)**, the execution of the Inventory Job has been successfully completed.

**Disabled**, an inventory analysis is not executed. The object is recorded in the Enterprise View Repository as an object for reference only.

The Overview tabsheet in the Enterprise View Builder shows the Object Qualifications Distribution by Analysis Status after execution of the Inventory Job.

### 8.4. When to Rerun Inventory Job

You must rerun the Inventory Job in the following circumstances:

1. There are errors during the execution of this job. Such errors can be:
   - Wrong directory paths
   - No write access to the directories
- Not enough space on the disk for writing

The following steps have to be performed:

1. Run **Inventory Deletion Job without Delete Structure** to delete all the information from the Enterprise View Repository and Database tables without the structure of the selected organization view.
2. Correct the errors.
3. Run Inventory Job again.

**II. There are missing sources and incorrect or missing extension assignments**

Usually this happens when the Inventory Probe Job has not been executed before executing Inventory Job. In this case, all problems concerning wrong or undefined extensions and missing sources will appear after executing the Inventory Job.

In case of missing sources, you can follow the instructions described in the chapter **Rerun after Adding New Components** and rerun Inventory after Inventory Probe Job.

In case of incorrect assignment of extensions to sources or missing extensions, you must follow the next steps:

1. Run **Inventory Deletion Job without Delete Structure** to delete all the information from the Enterprise View Repository and Database tables without the structure of the selected organization view.
2. Correct the wrong extensions or add the missing extensions to the Enterprise View Knowledge database.
3. Run the Inventory Probe Job and check the results (advisable).
4. Run Inventory Job again and check the results.
9. Starting Analysis Job

9.1. Introduction

The analysis phase is the most important phase of the product. During this phase, Enterprise View analyzes the sources in the indicated organization view based on their extensions, Enterprise View system, class, type, language definitions, and the type of analysis. The type of analysis (basic, extended, or disabled) can be shown using Environment Settings, the Work with Codes function.

During the analysis, Enterprise View:

- Identifies the definitions of the areas with the corresponding fields when possible.
- Identifies the relationships between objects.
- Identifies the dynamic links between the objects if there are dynamic calls in the source code.
- Calculates metrics.

All the information that is found is stored in the Enterprise View Repository and Tables and can be viewed by using client modules such as Enterprise View Builder or Enterprise View Tech.

The analysis can be run at any level of the technical view: package, application, application version, and component. The analysis process works within the package, so it is important that all the components necessary for processing the user’s environment have been included in the same package.

If the analysis is run at the package level, Enterprise View automatically processes the sources in order: It first stores the information about files and relationships between files and objects, and then handles include books and programs that need the information already loaded in order to be able to calculate metrics properly and provide a complete picture of their relationship.

If you do not start the analysis at the package level, it will run in the order expected by the Enterprise View product:

1. Database definitions (DDL, DBD)
2. Resource definitions (CICS CDS, IMS Stage 1)
3. Maps
4. JCLs
5. Include books
6. Programs

Following this order, Enterprise View will generate unresolved objects only if some software parts have not been included in the environment or have the wrong extensions.

At the end of the analysis phase, all the information collected in the Repository and Database tables will be available to view. Using the client solutions:

- Enterprise View Builder
- Enterprise View Tech
- Enterprise View Dynamic Inventory
- Enterprise View Impact Analysis

you can see the data from different points of view. For further details, refer to the reference guide of each Enterprise View client solution.
This job can be repeated as necessary. For more information, see the chapter When to Rerun Analysis Job.

9.2. How to run the Analysis Job

There are two ways to run the Analysis Job. In both cases, Job Wizard menus will start if you work with the newer versions of the Enterprise View Navigator.

1. Using the ‘Work with Jobs’ menu:

   - Click on the Work with Jobs icon and select the New button. If the Job Creation Wizard menu appears, go to the chapter Starting Host Jobs. Otherwise, select Analysis from the pop-up window and press the OK button.

   - The window showing incomplete fields will appear on the screen.
   - Select Advanced Parameters. The following window with a warning message will appear. You can change these settings or let Enterprise View do it for you.

   - Fill in the fields as described in the chapter “Work with Jobs” in the “Common Features” reference guide. The Analysis Job with the filled-in fields is ready for execution.
   - The ‘Organization View’ box enables Package, Application, Versioned Application or Component to be selected for analysis.
2. Using the Organization View Tree:

- Select a package, application, application version or component in the organization view tree.
- Right-click on the selected package, application, application version or component, choose the Job Submission item and select Submit Analysis Job from the pop-up menu as shown in the following picture. If the Job Creation Wizard menu appears, go to the chapter Starting Host Jobs.

In both cases, the ‘Work with Jobs’ window will appear with the information about the selected item from the Organization View Tree.

- Check ‘Update Trend’ if you are interested in the previous status and the movement of the metrics of the objects.
- Start a KbXDaemon session on the server machine.
- Press the OK button on the Work with Jobs window to submit the job for execution.
9.3. Analysis Job Results

After finishing the Analysis Job, you must verify that everything has been processed properly, see if errors have been found or if there are problems that require the user’s intervention. The Analysis Status field in Enterprise View Builder and Enterprise View Tech shows the results after execution of the Analysis Job for every object in the selected organization view. The analysis can be:

- **Basic** - information about the object (name, description, system, class, type, language, number of objects, number of lines) is recorded in the Enterprise View Database and in the Repository.
- **Extended** - the information from the basic analysis plus information about attributes, relations, dynamic links and metrics is recorded in the Enterprise View Database and in the Repository.
- **Disabled** - no analysis is performed. The objects are recorded in the Enterprise View Database and in the Repository as objects for relations only.

Once the analysis job has been run, all the sources should have the analysis status:

- **Done (Basic)**
- **Done (Extended)**
- **Done (with warnings, errors, timeout, not parsed).**

When the analysis status is:

- **Done (Basic)** - the basic analysis has been successfully completed.
- **Done (Extended)** - the extended analysis has been successfully completed.

The checks to be performed concern:

- **Unresolved Objects**
- **Sources with analysis status Done (with warnings)**
- **Sources with analysis status Done (with errors)**
- **Sources with analysis status Done (timeout)**
- **Sources with analysis status Done (not parsed)**

When you check the results after finishing the Analysis Job, take the following considerations into account:

- The level at which the analysis has been run: package, application, application version or component.
- The ‘right’ order of processing the analysis of the sources when the analysis is not performed at package level.
- The objects with analysis status Done (with warnings, errors, timeout and not parsed).

A quick way to verify the status of the analysis is to open Enterprise View Builder and select the Overview tabsheet, the Objects Qualification Distribution by Analysis Status section. The objects classified by analysis status are displayed below.
The All Objects tabsheet shows the Analysis Status of all objects in the selected organization view. You can see the source code of selected objects by clicking Show Source from the pop-up menu as shown in the next picture.
9.3.1. Unresolved Objects

One of the main features of the Analysis Job is the tracking of relationships between objects. If during the analysis of a source, Enterprise View does not find in the repository an object used or called by the source itself, it stores the missing object as an unresolved object. The unresolved object can be a program, include book, file, IMS or CICS resource or DL/I database definition. In the examples below, some cases of missing objects are described.

**Example 1**
Program A calls Program B, the source of Program A is in the Enterprise View Repository (belongs to the package), the source of Program B has not been included in the package. When Enterprise View analyzes the source of Program A and finds that there is a call to Program B but does not find program B in the repository, it stores Program B as an unresolved object.

**Example 2**
Program A uses File C, the source of Program A is in the Enterprise View Repository (belongs to the package) but no definition for File C is included in the package, for example: JCL file definition, DB2 DDL table definition, DL/I DBD PSB or segment definition etc.

When Enterprise View analyzes the source of Program A and finds that this source uses File C, but does not find File C in the repository, it stores File C as an unresolved object.

**Example 3**
Program A uses Include Book D. The source of Program A is in the Enterprise View Repository (belongs to the package) but the source of Include Book D has not been included in the package.

When Enterprise View analyzes the source of Program A and finds that there is a reference to Include Book D but does not find Include Book D in the repository, it stores Include Book D as an unresolved object.

The presence of an unresolved object usually means that:

- Not all objects have been included in your application.
- Not all objects have been transferred correctly and included in the organization view.
- If the analysis has been performed at application, application version, or component level, you have not run the analysis in the order expected by Enterprise View.
- Enterprise View cannot calculate some metrics properly. For example, if the missing object is an include book, this may affect the metrics calculated for the program that uses this include book. The metrics can be Halstead, MI, etc.
- Enterprise View will not create relationships between these objects.

**NOTE:** The list of unresolved objects is important as a means of locating the missing objects.

Usually, your applications contain thousands of objects and sometimes it is not easy to immediately identify all libraries and folders that contain the sources of the application or to have all the sources available at the time of the inventory phase.
Using the list of unresolved objects it is possible to:

- Check the number of missing objects.
- Simplify the search for sources in the development environment

### 9.3.1.1. How to Manage Unresolved Objects

The most common causes of unresolved objects and how to manage them are described in this section.

#### Missing application programs

These are software parts that have not been included in the package because their sources were not available or because you forgot to put them in the organization view when it was generated.

**How to solve the problem:**
Add the missing software parts to the package using the Define Technical View functions and rerun the Analysis job following the necessary steps as explained in the chapter *When to Rerun Analysis Job*.

#### Sources analyzed in the wrong order

As explained in the chapter *Introduction*, the analysis can be run at any level of the organization view: package, application, application version, or component. If the analysis is not run at the package level and the sources are analyzed in the wrong order, unresolved objects might be created.

**How to solve the problem:**
Rerun the analysis of the components in the right order following the steps in the chapter *When to Rerun the Analysis Job*.

#### System utility sources not available

There are system utility programs or database interface programs used by the application programs whose source is never available, for example:

- System utilities: IEHMOVE, ICEGENER, IEBGENER, IEFBR14, IEUPDATE, IEBCOPY, IEHLIST, IEHPROGM, EZCOPY, SORT, IDCAMS
- DB2 loaders: IKJEFT01, IKJEFT1A
- IMS loaders: DFSRRC00, DFSURGU0
- IMSDB2 loader: DSNMTV01

Usually they appear as unresolved objects.

**How to solve the problem:**
This problem can be solved by entering in the LOADERS section of the configuration file *JclConfig.cfg* the names of all loaders that the application uses, as shown in the next example. The *JclConfig.cfg* file is found in the *Job* directory on the Enterprise View server machine.
The LOADERS section of file JclConfig.cfg:

[LOADERS]
[LOADERS]
NAME = 'IKJEFT01','IKJEFT1A'
TYPE = DB2

[LOADERS]
TYPE = IMS
NAME = 'DFSRRC00'

[LOADERS]
TYPE = IMSDB2
IMS_NAME = 'DFSRRC00'
DB2_NAME = 'DSNMTV01'

[LOADERS]
TYPE = IDC
NAME = 'IDCAMS'

**9.3.1.2. How to identify ‘Unresolved Objects’**

If there are unresolved objects after finishing the Analysis Job for the selected organization view, they can be viewed in the Enterprise View Tech client solution.

- Open Enterprise View Tech.
- Select the Unresolved tabsheet. It contains the list of all unresolved objects for the selected organization view.
- Select one object from the list and right-click. The dropdown menu will appear.
- Using the Show Source function from the dropdown menu, it is possible to find out if the source has been included in the organization view or not.
- Using the Browse Object function from the dropdown menu, it is possible to follow the reference and to display the line where the item was involved.
- Using the Reference section it is possible to see all the characteristics of this reference and the part from which the unresolved object originates.

In the next picture, the Show source function will show the source of the object (in this case JCL file) and the line at which the unresolved card’s object is called.
9.3.2. Analysis status Done (with warnings)

After completing the analysis of the sources, Enterprise View finds warning errors (usually unsupported or unrecognized syntax) that may affect the reliability of metric calculations or relationships retrieval.

How to solve the problem:
1. Submit Unknown Syntax Job. This job reports the names of all objects that contain unsupported or unrecognized syntax. For more information see the chapter Starting Unknown Syntax Job.
2. Select the tabsheet ‘Unsupported syntax’ in the Enterprise View Builder module and verify for each object the type of syntax Enterprise View does not understand.
3. Correct the problem and rerun the necessary steps as explained in the chapter When to Rerun the Analysis Job.

9.3.3. Analysis status Done (with errors)

When the analysis finishes with Done (with errors) this means that the analysis has not been completed because of a serious problem found in the job. This may be a ‘Segmentation fault’ error, a wrong type of source (for example the source is in binary format, or contains unknown syntax), a wrong extension, or some other serious error. The metrics are not calculated at all or are unreliable. Created relationships are unreliable too.

How to solve the problem:
Verify that the sources marked with this error contain the correct type of code and extension, and rerun the analysis following the instructions in the chapter When to rerun the Analysis Job. If the problem persists, contact Micro Focus SupportLine.
**9.3.4. Analysis status Done (timeout)**

This error appears after the execution of Analysis Job and means that the analysis of the object has not been completed because no answer was provided within the maximum analysis time specified at the time of the installation. Metric calculations and/or relationships retrieval may be unreliable.

**How to solve the problem:**
Verify that the machine meets the minimum requirements described in the document “Hardware and Software Requirements”. Ask your system administrator to check that the message was not due to:
- Unavailable resources on the machine
- Competitor jobs on the same database.

If the problem persists, contact Micro Focus SupportLine.

**9.3.5. Analysis status Done (not parsed)**

This error appears after submission of an Analysis Job and means that the analysis of the objects has not been completed due to a parser fault.

**How to solve the problem:**
Contact Micro Focus SupportLine.

**9.3.6. Sources with status To Do (Extended) after analysis job**

Some sources still have a status of To Do (Extended) after execution of an analysis job. Possible reasons for this are:

**I. Messages for:**

- Wrong extensions for the type of source.
- Unsupported syntax lines in the source.

In this case, the follow steps have to be performed:
1. Correct the extension for the type of source.
2. Correct the syntax in the source
3. Run **Inventory Deletion Job without Delete Structure** to delete all objects with detailed information about them from the Enterprise View Repository and Database tables for the selected package.
4. Run Inventory job for the package.
5. Run the Analysis job for the package.

**II. Analysis job does not run for some components**

If the analysis is not executed at a package level, some components have been missed for analysis. In this case, run the Analysis job for the components which were never analyzed taking in account the order of the analysis process as described in the chapter **Introduction**. If the analysis has not been performed in the right order, rerun the analysis follow the instructions in the chapter **When to rerun the Analysis Job**.

**III. Analysis job has stopped for any causes**
Rerun the analysis following the instructions in the chapter When to rerun the Analysis Job.

9.4. When to Rerun the Analysis Job

The Analysis Job must be rerun in the following situations:

**I. When there are errors during the execution of this job**

Cases with error messages can be:
- Sources with analysis status Done (with warnings)
- Sources with analysis status Done (with errors)
- Sources with analysis status Done (timeout)
- Sources with analysis status Done (not parsed)
- Sources analyzed in a wrong order
- Sources with unknown syntax
- Other errors (no enough space for writing, database problems or analysis job has stopped for any causes)

The follow steps have to be performed:

1. Run **Inventory Deletion Job without Delete Structure** to delete all objects with the detailed information about them from the Enterprise View Repository and Database tables for the selected package without the structure of the organization view.
2. Correct the errors.
3. Run Inventory Job for the same package again.
4. Run Analysis Job for the same package. If the analysis is not executed at the package level, keep the right order for analysis of the component.

**II. When there are unresolved objects (missing sources)**

If for some reason, unresolved objects different from the loaders or system utility programs still exist, you need to add them to the package. There are two ways to add missing sources to the package and rerun the Analysis Job:

- **By executing Inventory Deletion Job with delete structure**
  The follow steps have to be performed:
  1. Run **Inventory Deletion Job with Delete Structure** to delete all objects with the detailed information about them from the Enterprise View Repository and Database tables for the selected package with the structure of the organization view.
  2. Transfer the missing sources to the new folders and check the file extensions.
  3. Define the Technical Organization View including the new components.
4. Run Inventory Probe Job and check the results (advisable).
5. Run Inventory Job and check the results.
6. Run Analysis Job and check the results.

☐ By executing Inventory Deletion Job without delete structure

The following steps have to be performed:

1. Run **Inventory Deletion Job without Delete Structure** to delete all objects with the detailed information about them from the Enterprise View Repository and Database tables for the selected package without the structure of the organization view.
2. Transfer the missing sources to the existing folders and check the file extensions.
3. Run Inventory Probe Job and check the results (advisable).
4. Run Inventory Job and check the results.
5. Run Analysis Job and check the results.

**Important:** After executing Analysis Job, Inventory Job can be rerun only after executing the Inventory Deletion Job for the selected organization view.
10. Starting Life Cycle Job

10.1. Introduction

The Life Cycle job is performed after successfully finishing the first part of the analysis: inventory and analysis phases for the selected organization view. It can be repeated in cases of errors, unresolved objects (missing sources), or wrong extensions. This job can be executed at package level only. The Life Cycle job tracks if there are changes in the application (for example new sources, deleted sources or sources with changed program code) and it updates the information in the Enterprise View Repository and Database tables. This job performs an inventory and analysis function:

- The files are compared file by file.
- The program code in the sources is compared and the changes are updated.
- All new are written to the Enterprise View Repository and Database tables.
- The information about the files with changed program code and deleted files is updated.
- Analysis of all the (new, changed and unchanged) sources is executed – the metrics and relations are updated.

In order to update the analysis results (attributes, relations, metrics), when the sources in the applications are changed frequently, this job can be scheduled for execution daily using the Enterprise View Agent.

10.2. How to run Life Cycle Job

There are two ways of running Life Cycle Job:

1. Using the ‘Work with Jobs’ menu:
   - Click on the Work with Jobs icon and select the New button.
   - Select Life Cycle from the drop down window and press the OK button. If the Job Creation Wizard menu appears, go to the chapter Starting Host Jobs. Select Life Cycle from the drop down menu and press the OK button.
2. Using an Organization View Tree:

Right-click the package in the Organization View Tree and choose the Job Submission item and select Submit Life Cycle Job from the drop down menus. If the Job Creation Wizard menu appears, go to the chapter **Starting Host Jobs**.

In both cases, the Work with Jobs window will appear with information about the selected package.
- Start a KbXDaemon session on the server machine.
- Press the OK button on the Work with Jobs window to submit the job.
10.3. Understanding Life Cycle Status Results

The results of the Life Cycle Job can be viewed in the Enterprise View Builder and Enterprise View Tech. The Status field in the tabsheets for every class of object (programs, copy/includes, JCLs, transactions, maps, etc) shows the changes after execution of the Life Cycle Job. The status can be:

- New – a new object has been added in the Enterprise View Repository and Database tables.
- Changed – the program code in the existing object has been changed.
- Unchanged – the program code in an existing object has not been changed.
11. Starting the Unknown Syntax Job

11.1. Introduction

The Unknown Syntax Job must be started after execution of the Analysis Job. It works at every level of the technical organization view: package, application, application version or component. All objects in the selected organization view with unknown syntax are recorded in the `kbAIM_UnknownSyntax.txt` file in the Job directory. Unknown syntax is syntax that is wrong or not supported by Enterprise View. You can see the programming languages that Enterprise View supports in the Environment Settings of the Enterprise View Configurator. If there are objects with unknown syntax, the information about these objects is written to the Enterprise View Database.

11.2. How to run the Unknown Syntax Job

This job can be run in the same way as the Life Cycle Job but you submit Unknown Syntax Job instead of Life Cycle Job.

11.3. Understanding Unknown Syntax Results

A list of all objects that contain unknown syntax lines can be viewed using the Unsupported Syntax tabsheet of the Enterprise View Builder module. The following information is reported:

- **Name** - the name of the object that contains unknown syntax.
- **Location** - the number of the line with unknown syntax.
- **Unsupported Syntax Line** – the text identified as unknown syntax.

The number of the line with the Unrecognized Syntax for every object appears when you click the object containing the syntax in the Enterprise View Tech module.

The possible reasons can be:

- **Wrong extensions for the type of source**
  
  **How to solve the problem:**
  
  Correct the extensions and rerun the job.

- **Wrong syntax in source code**
  
  **How to solve the problem:**
  
  Correct the syntax and rerun the job.

- **Binary or other unspecified text in the source**
  
  **How to solve the problem:**
  
  Exclude the object with this type of source from the organization view and rerun the job.
12. Starting the Inventory Deletion Job

12.1. Introduction

This job deletes objects assigned to organization views, their relations, and metrics. Depending on the delete structure flag it is also possible to delete the organization view structure.

The Inventory Deletion Job can be started at every level of the technical organization view: package, application, application version or component. This job performs the following operations:

- Deletes all objects and the detailed information about them from the Enterprise View Database tables and Repository, but not the structure of the organization view if the Delete Structure box is not checked.
- If the Delete Structure box is checked, the job deletes all objects and the detailed information about them from the Enterprise View Database tables and Repository, and also deletes the structure of the selected level of the organization view: package, application, application version or component.

12.2. How to run the Inventory Deletion Job

This job can be run using the Organization View Tree. In the newer Enterprise View versions this job can also be started using the ‘Work with Jobs’ menu. Select the desired level from the Organization view tree (package, application, application version or component), right-click and choose the Job Submission item and select Submit Inventory Deletion Job from the drop down menus. If the Job Creation Wizard menu appears, go to the chapter Starting Host Jobs. Otherwise you can edit the job parameters in Work With Jobs.

Check the Delete Structure box, if required. Start a kbXDaemon session on the server machine. Click the OK or Apply button to submit the job.

**WARNING:** We recommend you do not execute an Inventory deletion job during the second phase of the analysis process (Life Cycle process) when the Trend box is checked during the analysis, or the trend information (previous status and the movement of the metrics of the objects) will be lost.
13. Starting the HTML Export Job

This job can be used for exporting organization view structures with some statistics in HTML format. HTML export can be performed at every level of the technical organization view: package, application, application version or component. The job can be started using the 'Work with Jobs' menu or using the 'Documentation Export' of the Solutions module. Checking Refresh Existing HTML Export box, you can see for which organization views or their levels in the package, HTML documentation export already exists. If you start a job for a package that already has HTML exported documentation, only the information for updated objects will be added to the existing exported documentation.

Three general types of information are reported:

- Overview information
- Object list information
- Details object information.

If you open index.html, you will see detailed information about all linked objects, their attributes, and relations. Depending on some options it is also possible to export object source code.
14. Starting the End Daemon Job

The End Daemon Job is used to halt execution of the started kbXDaemon. This job performs the following operations:

- Checks if there is a job running in kbXDaemon: for example Inventory Job, Analysis Job, Inventory Deletion Job, Life Cycle Job, and so on.
- Sends notification to the kbXDaemon to stop when the current job has finished its work.
- If no job is running, End Daemon Job stops the kbXDaemon immediately.

This job can be started only by using the Job Creation Wizard menus.
15. Analyzing DL/I sources

15.1. IMS DB DL/I General Considerations

The IMS DL/I database uses a hierarchical model for storing data. In this model, the individual entity types are implemented as segments in a hierarchical structure. The hierarchical structure is determined by the designer of the database, based on the relationship between the entities and the access paths required by the applications. The basic element of a hierarchical data structure is the parent/child relationship between segments of data. Compared to the relational model used by DB2, a DL/I database is approximately equivalent to a table. There are a number of different IMS access methods used to organize and store the data segments and records. The most used DL/I database organizations are:

- Hierarchical Direct Access Method - HDAM
- Hierarchical Index Direct Access Method - HIDAM
- Simple Hierarchical Index Sequential Access Method - SHISAM
- Hierarchical Index Sequential Access Method - HISAM
- Generalized Sequential Access Method - GSAM
- Data Entry Database - DEDB

The physical DL/I database is created using:

- VSAM (Virtual Storage Access Method) files
- ISAM (Index Sequential Access Method) files
- OSAM (Overflow Sequential Access Method) files

Components of the IMS DL/I database from a physical and program point of view are:

- DBD - Database Descriptor Block
- PSB - Program Specification Block
- PCB - Program Communication Block

15.2. Sources containing DL/I Database Definitions

15.2.1. Introduction

For physical and logical description of the IMS DL/I databases, the following languages are used:

1. Database Description Language (DBD).
2. VSAM Delete/Define Language for VSAM files.
3. JCL DD statement definitions for VSAM, ISAM and OSAM files.

During analysis of the sources containing DL/I, VSAM Delete/Define definitions, or JCL DD statement definitions, the Enterprise View Parsers scan the sources and load the Enterprise View Repository information for the files and database definitions as well as for the following measures:

- Lines of Code - LOC
- Source Lines of Code - SLOC
- Comment Lines of Code - CLOC
- Change Traffic
Enterprise View classifies all DL/I database files as Physical. It does not examine the ACCESS clause: LOGICAL, PHYSICAL, INDEX, or GSAM.

The following class-type-language classification for DL/I physical files is contained in the Enterprise View Knowledge database:

- DL/I File - Data Set
- DL/I File - Segment
- DL/I File - DBD

The analysis type of these files is set to Disabled in the Enterprise View Knowledge Database and they are not therefore analyzed. The file names are included in the Repository and in the Enterprise View tables for inventory purposes only.

**NOTE:** Analysis of the database definitions must be carried out like the first step of the analysis (when the analysis is not at package level). Otherwise Enterprise View creates numerous unresolved objects in the subsequent phases of the analysis (i.e. during the analysis of the programs and copy/include books).

### 15.2.2. DL/I Descriptor Language

For a description of the DL/I components (DBD, PSB, and PCB), DL/I database descriptor language is used. Application programs can communicate with databases without being aware of the physical location of the data they possess.

The Database Description Block (DBD) describes the content and hierarchic structure of the physical or logical database.

The Program Specification Block (PSB) specifies the database segments that application programs can access and the functions they can perform on the data, such as read-only, update, or delete. PSBs are composed of one or more program control blocks (PCBs). Analysis of the program specification blocks is not performed by Enterprise View.

The Program Communication Block (PCB) defines the logical structure of the database from an application program’s view. Analysis of the program communication blocks and their hierarchy is not performed by Enterprise View.
Database Description Language (DBD)

The sources containing DBD elements must have an extension `.dbd` or `.DBD` to be properly analyzed by the Enterprise View parser. The analysis type for DBD sources is set to Extended in the Enterprise View Knowledge database. During the DBD analysis, the Enterprise View parser creates:

- The DBDs list
- The SEGMENTS list
- The SEGMENTS hierarchy (parent/child segments).

The relation types that the Enterprise View parser tracks and loads are:

- Create DBD
- Create Data
- Create Segment
- Hierarchy DBD
- Hierarchy Data
- Hierarchy Segment

The status of database definitions after the analysis phase can be viewed using the Database Definition tabsheet of the Enterprise View Builder or Enterprise View Tech.

The file `DI21PART.dbd` contains the definition for the `DI21PART` database. This database contains five segments: `PARTROOT`, `STANINFO`, `STOKSTAT`, `CYCCOUNT`, and `BACKORDR`.

* TITLE 'DBFSAMD4 - FP SAMPLE DBD FOR CUSTOMER ACCT. - HDAM'
  DBD NAME=`DI21PART`,ACCESS=(HISAM,VSAM)
  DATASET DD1=DI21PART,DEVICE=3380,OVFLW=DI21PARO,
  SIZE=(2048,2048),RECORD=(678,678),RECORD=(678,678)
  SEGM NAME=`PARTROOT`,PARENT=0,BYTES=50,FREQ=250
  FIELD NAME=(PARTKEY,SEQ),TYPE=C,BYTES=17,START=1
  SEGM NAME=`STANINFO`,PARENT=`PARTROOT`,BYTES=85,FREQ=1
  FIELD NAME=(STANKEY,SEQ),TYPE=C,BYTES=2,START=1
  SEGM NAME=`STOKSTAT`,PARENT=`PARTROOT`,BYTES=160,FREQ=2
The user can see the DBD/SEGMENT relations using the ‘Browse Object References’ function of the Enterprise View Tech module. The following screen shows the ‘Create DBD’ and ‘Create DL/I Segments’ relation types.

15.2.3. VSAM Delete/Define File Statements

A VSAM file is known as a cluster. VSAM clusters are managed by a utility called IDCAMS. VSAM is a Virtual Storage Access Method that allows:

- sequential access to files either by a key or by sequence;
- direct access by key, relative byte address, or relative record number.

For description of the VSAM clusters, IDCAMS utility control statements are used.

**VSAM Delete/Define File**

The file containing the source code of the VSAM Delete/Define File statements must have an extension `.ddef` or `.DDEF` to be properly analyzed by the Enterprise View Parsers. Usually VSAM statements are located in JCL SYSIN DD * statements or are written as members of the partitioned dataset that is included in input DD statements during execution of the IDCAMS step.

The Enterprise View Parser extracts and loads the Enterprise View Repository as a physical file (VSAM Cluster and Path definitions) and as a logical file VSAM Alternate Index definitions.
**15.2.4. JCL Physical Files**

Datasets that belong to DL/I databases are specified with JCL DD statements. If, during analysis of the JCL sources, Enterprise View finds descriptions of the datasets, these files will be recorded as physical files in the Enterprise View Repository. The analysis type of the JCL files (ISAM, OSAM and VSAM) is set to Disabled in the Enterprise View Knowledge database and so is not analyzed. The file names are included in the Repository for inventory purposes only. Enterprise View gives information about these files after execution of the analysis job. The status of the physical files after the analysis phase can be viewed using the Files tabsheet of the Enterprise View Builder or Enterprise View Tech.

For information about the Enterprise View analysis of the JCLs, JCL Cards and Catalogued Procedures, Copy/Includes, Programs, Transactions and the other components of the applications running in IMS and CICS environment, see the “IMS Process Guide” and “CICS Process Guide”.

DLI Process Guide
16. DL/I Program Language Interface

16.1. Introductions

Application programs can access DL/I databases using the following program language interfaces:

- Command level language interface - EXEC DLI
- Call level language interfaces:
  - CALL CEETDLI
  - CALL AIBTDLI
  - CALL xxxxTDLI

Enterprise View supports the following program languages in the IMS and CICS environment: Assembler, C/C++, COBOL, COBOL Delta Macro and PL/I.

16.2. EXEC DLI Command Interface

The EXEC DLI commands can be used to read and update DL/I databases with IMS batch programs, IMS online programs (running DBCTL or DB/DC), or CICS programs using DBCTL (DB control).

The common format of the command is: EXEC DLI function USING options

The format of the command is the same for users of COBOL, COBOL Delta Macro, PL/I, Assembler, and C/C++ languages.

For information on how Enterprise View analyzes EXEC DL/I commands, see the chapter Enterprise View Analysis of DL/I Commands.

16.3. CALL DLI Language Interface

The CALL level interface is used in programming languages supported by IMS - Assembler, C/C++ language, COBOL, Pascal, and PL/I. The programmer must CALL the DL/I language interface to initiate the functions specified with the DL/I calls. IMS offers several interfaces for DL/I calls:

- CEETDLI - a language-independent interface. IMS does not need to know the language of the calling program. This interface is not analyzed by Enterprise View.
- AIBTDLI - a non-specific language interface for all supported languages. This interface is not analyzed by Enterprise View.
- xxxxTDLI - language-specific interfaces for all supported languages. The language of the calling program is determined from the entry name that is specified in the CALL statement. The format of the CALL command for the program languages supported by Enterprise View is:
  - Assembler - CALL ASMTDLI,(function,db pcb,i/o area,ssa),VL
  - C/C++ - ...
  - COBOL - CALL 'CBLTDLI' USING function,db pcb,i/o area,ssa.
  - PL/I - CALL PLITDLI (parmcount,function,db pcb,i/o area,ssa);

For information on how Enterprise View analyzes CALL DL/I commands, see the chapter Enterprise View Analysis of DL/I Commands.
17. Enterprise View Analysis of DLI Commands

The following chapters describe how Enterprise View analyzes the DL/I commands in IMS and CICS environments.

17.1. Sources with Basic Analysis

Enterprise View performs basic analysis for:
- IMS Batch Assembler programs
- IMS Batch C/C++ programs
- IMS/TP Assembler programs
- IMS/TP C/C++ programs
- CICS Assembler programs
- CICS C/C++ programs

Analysis of the DL/I commands (EXEC DLI and CALL xxxxTDLI) for these program languages is not performed.

17.2. Sources with Extended Analysis

Enterprise View performs extended analysis for:
- IMS Batch COBOL and COBOL Delta Macro programs
- IMS Batch PL/I programs
- IMS/TP COBOL and COBOL Delta Macro programs
- IMS/TP PL/I programs
- CICS COBOL and COBOL Delta Macro programs
- CICS PL/I programs

EXEC DL/I commands

The common format of the command is:
EXEC DLI function USING options

where: function is specified DL/I function code (for example: GN, GU, DLET, INSR …)
options are specified keywords and parameters that describe segments, PCBs, SSAs and other parameters.

Analysis of the EXEC DL/I commands is executed only for COBOL (respectively COBOL Delta Macro) sources.

NOTE: In PL/I sources, analysis of EXEC DL/I commands is not performed.

CALL DL/I commands

The format of the CALL DL/I command is:

For PL/I: CALL PLITDLI (parmcount.function.db pcb,i/o area,ssa);
For COBOL: CALL 'CBLTDLI' USING \texttt{function,db pcb,i/o area,ssa}.

Where: \texttt{parmcount} specifies the number of the parameters
\texttt{function} specifies the DL/I function code (for example: GN, GU, DLET, INSR ...)
\texttt{db pcb} specifies the address of the database PCB
\texttt{i/o area} specifies the address of the I/O area
\texttt{ssa} specifies the address of the segment search argument. Up to 15 SSAs can be specified.

Analysis of the CALL xxxxTDLI commands is performed for COBOL (respectively COBOL Delta Macro) and PL/I sources by Enterprise View.

During analysis of the commands Enterprise View:

- Classifies the DL/I Calls type according to the total number of parameters for the CALL and the value of the first of these parameters that represents the DL/I Function Code.
- Assumes the Function Code to be the first parameter of the CALL in the COBOL sources and the second parameter in the PL/I sources.
- Handles Function Codes that generate a significant relationship with an object.
- When multiple SSAs are used for a Call, the relationship Program/Segment is created only for the last SSA parameter used in the call.
- Does not take into account segments having the same name and defined in more than one DBD.

After analysis, the following relations are created by Enterprise View:

- Program/Segments
- Program/Generic Object

The relation types can be: read, update, insert, delete.

In the next table, Function Codes that generate a relationship Program/Segment for Database Management are shown.

<table>
<thead>
<tr>
<th>Function Code</th>
<th>Relation type</th>
<th>Number of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLET (Delete )</td>
<td>delete</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GHN (Get hold next )</td>
<td>read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GHNP (Get hold next in parent)</td>
<td>read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GHU (Get hold unique )</td>
<td>read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GN (Get next )</td>
<td>read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GNP (Get next in parent)</td>
<td>read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GU (Get unique )</td>
<td>read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>INSR (Insert )</td>
<td>insert</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>REPL (Replace )</td>
<td>update</td>
<td>&gt;= 4</td>
</tr>
</tbody>
</table>
In the next table, the Function Codes that generate a relationship Program/Generic Object for Transaction Management are shown.

<table>
<thead>
<tr>
<th>Function Code</th>
<th>Relation Type</th>
<th>Number of the Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTH</td>
<td>Read</td>
<td>= 3</td>
</tr>
<tr>
<td>CHNG</td>
<td>Read</td>
<td>= 3</td>
</tr>
<tr>
<td>CMD</td>
<td>Read</td>
<td>= 3</td>
</tr>
<tr>
<td>DLET</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>FLD</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GCMD</td>
<td>Read</td>
<td>= 3</td>
</tr>
<tr>
<td>GHN</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GHNP</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GHU</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GN</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GNP</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>GU</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>ISRT</td>
<td>Insert</td>
<td>= 3</td>
</tr>
<tr>
<td>ISRT</td>
<td>Insert</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>POS</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>PURG</td>
<td>Read</td>
<td>= 3</td>
</tr>
<tr>
<td>PURG</td>
<td>Read</td>
<td>&gt; = 4</td>
</tr>
<tr>
<td>REPL</td>
<td>Read</td>
<td>&gt;= 4</td>
</tr>
<tr>
<td>SETO</td>
<td>Read</td>
<td>= 3</td>
</tr>
</tbody>
</table>