The YGrep Search Engine is Copyright (c) 1992-2004 Yves Roumazeilles. All other trademarks are the property of their respective owners.

No part of this publication, with the exception of the software product user documentation contained on a CD-ROM, may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine-readable form without prior written consent of Micro Focus (IP) Ltd. Contact your Micro Focus 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 Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.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. PREPARATORY ACTIVITIES .................................................. 9
   2.1. PREREQUISITES ....................................................... 9
   2.2. UNDERSTANDING ANALYSIS PROCESS PHASES ................. 9
3. ENTERPRISE VIEW PROCESS OVERVIEW ............................... 10
4. PROVIDING SOURCES TO ENTERPRISE VIEW SERVER MACHINE .......... 13
   4.1. INTRODUCTION ....................................................... 13
   4.2. ENTERPRISE VIEW EXTENSIONS FOR IMS BATCH ENVIRONMENT .... 15
   4.3. ENTERPRISE VIEW EXTENSIONS FOR IMS ONLINE ENVIRONMENT ... 15
   4.4. MISSING SOURCES .................................................. 16
      4.4.1. Missing Programs or Transactions ............................ 16
      4.4.2. Missing Copy/Include Books .................................. 16
      4.4.3. Missing JCLs and JCL Catalogued Procedures ............... 17
      4.4.4. Missing IMS Stage One ....................................... 17
5. DEFINING AN ORGANIZATION VIEW TREE .............................. 19
6. STARTING HOST JOBS ....................................................... 21
7. STARTING INVENTORY PROBE JOB ....................................... 25
   7.1. INTRODUCTION ....................................................... 25
   7.2. HOW TO RUN INVENTORY PROBE JOB .............................. 25
   7.3. INVENTORY PROBE JOB RESULTS UNDERSTANDING .............. 27
   7.4. WHEN TO RERUN INVENTORY PROBE JOB ......................... 29
      7.4.1. Rerun after Adding New Components ......................... 29
      7.4.2. Rerun after Adding New Extensions .......................... 29
8. STARTING INVENTORY JOB ................................................ 31
   8.1. INTRODUCTION ....................................................... 31
   8.2. HOW TO RUN INVENTORY JOB ...................................... 31
   8.3. INVENTORY JOB RESULTS ......................................... 33
   8.4. WHEN TO RERUN INVENTORY JOB .................................. 35
9. STARTING ANALYSIS JOB ................................................ 37
   9.1. INTRODUCTION ....................................................... 37
   9.2. HOW TO RUN ANALYSIS JOB ...................................... 38
   9.3. ANALYSIS JOB RESULTS ......................................... 40
      9.3.1. Unresolved Objects .......................................... 42
   9.4. WHEN TO RERUN ANALYSIS JOB .................................. 47
10. STARTING LIFE CYCLE JOB ............................................... 49
   10.1. INTRODUCTION ...................................................... 49
   10.2. HOW TO RUN LIFE CYCLE JOB .................................... 49
   10.3. UNDERSTANDING LIFE CYCLE STATUS RESULTS .................. 51
11. STARTING UNKNOWN SYNTAX JOB ...................................... 53
1. About This Document

This document is a guide for Enterprise View users who work on a mainframe platform in the IMS environment. It describes how to identify sources in the different applications and analyze them following the Enterprise View requirements for the analysis process.

Applications working in the IMS environment can be written in different programming languages (Assembler, COBOL, PL/I, Java, Pascal, etc). They can have access to DB2 databases, IMS DB DL/I databases and different types of systems or user data sets. This document describes how to prepare applications written in different programming languages and working in different IMS environments (batch and online) for analysis, using Enterprise View.

You can read the information in this document from beginning to end to get a full understanding of the topics. Since the chapters are intended to be self-contained, advanced users can read them individually.
2. Preparatory Activities

2.1. Prerequisites

In order to be able to configure and analyze applications, the following software has to be installed beforehand:

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)
5. UNIX Agent (Optional)

You need the appropriate authorization (User Identifier and Password) to ensure:

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

For more information about the installation, administration and configuration of these products see:

- 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 analysis of the applications working in IMS environment. The overview represents the sequence that must be followed step by step for successful execution of the entire process. Depending on the messages and the different cases of analysis, some of the steps can be repeated.

This overview is valid for applications running in all IMS environments (batch and online) and developed with programming languages that Enterprise View supports: Assembler, C, C++, COBOL, COBOL Delta Macro, and PL/I.

The analysis process has two parts:

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

You must execute the Inventory and Analysis Jobs for a given defined organization view till successful completion of the primary analysis process. You can repeat the execution of Inventory Job ONLY after executing Inventory Deletion Job. The Life Cycle Job is executed when modifications have been made in the applications after the primary analysis process has been successfully completed.
3. Enterprise View Process Overview

Part I. Primary analysis process overview

- Transfer sources running in IMS environments to Enterprise View Server
- 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
- Add the Components
- Check if there are Unsupported Files and wrong extensions
  Work with File Extensions
  Yes
- Run Inventory Deletion Job with Delete Structure
  No
Part II. Post-analysis (Life Cycle) process overview

Transfer modified sources of the application to the Enterprise View Server machine in the corresponding components of the Organization View

Select an application (already successfully analyzed)

Run Life Cycle Job

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

Correct the errors

Check for Proposed Dynamic Links

Confirm Dynamic Links

Transfer modified sources of the application to the Enterprise View Server machine in the corresponding components of the Organization View

Client Modules:
Enterprise View Tech
Enterprise View Builder

Add missing sources

Yes

Yes

No

No

Yes
4. Providing Sources to the Enterprise View Server Machine

4.1. Introduction

The IMS applications can be created and tested directly in the mainframe environment (z/OS, OS/390, MVS). Using the Micro Focus Mainframe Express product, you can create IMS applications in the Windows environment (Windows 95/98, Windows 2000, and Windows NT 4.0) and test them on IMS/ESA platform. If the sources are not on the Enterprise View server machine they must be transferred to it in order to be analyzed.

The standard way to transfer sources is FTP (File Transfer Protocol). The source transfer can be performed by the Enterprise View MVS Agent Module when the applications are located on a mainframe platform. Any other transfer methods and protocols can be used in companies where Enterprise View MVS Agent is not installed.

The transferred sources can be located in one or in several folders. We strongly recommend you download the sources using the same folder organization as in 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 one and the same type of source code. For example, COBOL copybooks have to be in one folder, IMS/TP PL/I programs in another, JCL programs in a third, and so on. Additionally, each type of source must be transferred with an extension which conforms to the type of the source recognized in Enterprise View Knowledge Base. To simplify matters, a specific module called MVS Custom Agent can be installed to allow a quick configuration of the sources that will be transferred to the Micro Focus Enterprise View Server. This module allows partitioned selection and automatic extension assignment.

You can create and delete your own extensions using the Work with File Extensions menu in the Enterprise View Configuration Manager. This operation can be performed only by an Administrator of the product. The source extensions must conform to the new extensions.
The highlighted extensions in the tables below are the Enterprise View extensions. They are grouped depending on the environment in which the IMS applications are executed. The IMS applications can be executed in:

- Batch environment: IMS Batch
- Online environment: IMS Online

Extension assignment is mandatory, otherwise the sources will not be recognized.

NOTES:
1. Some programs have ‘general’ extensions: `.c`, `.C`; `.cbl` or `.CBL`; `.pli` or `.PLI`, and other programs 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. There is a benefit to using an extension linked to the environment, because it makes it easier to display the data in the client modules (i.e. Enterprise View Tech and Enterprise View Builder) because the Type column will show different values depending on the environment (i.e. IMS Batch or IMS Online). The Overview tabsheet will display the programs’ distribution by type and it will be easier to extract the programs using the Filter Active function in the Type column.

2. Extensions for Java and Pascal programming languages for OS/390 system cannot be created in the Enterprise View Knowledge Database because there are no Class-Type-Language qualifications for these languages in the Environment Settings of the Enterprise View Configuration Manager module for OS/390 system. For this reason, Java and Pascal program sources will not be discussed in this document.
4.2. Enterprise View Extensions for IMS Batch Environment

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

<table>
<thead>
<tr>
<th>Components</th>
<th>Description</th>
<th>Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programs</strong></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>.cppims, .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>.cdmims, .CDMIMS</td>
</tr>
<tr>
<td></td>
<td>IMS PL/I Programs</td>
<td>.pli, .pliims, .PLI, .PLIIMS</td>
</tr>
<tr>
<td><strong>JCL</strong></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><strong>Copy/Include</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 Books</td>
<td>.inc, .INC</td>
</tr>
<tr>
<td><strong>Files</strong></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>DB2 DDL</td>
<td>.ddidb2, .DDLDB2</td>
</tr>
<tr>
<td></td>
<td>VSAM Delete Define</td>
<td>.ddef, .DDEF</td>
</tr>
</tbody>
</table>

**Note:** If any of these extensions are missing in the Enterprise View Knowledge Database, you can create the same or other extensions using the Work with File Extensions menu in the Environment Settings part of Enterprise View Configuration Manager. This operation can be performed only by an Administrator of the product.

4.3. Enterprise View Extensions for IMS Online Environment

The extensions described in the following table are pre-defined extensions in the product. You can use them for applications containing IMS programs and sources running in an online environment.
<table>
<thead>
<tr>
<th>Components</th>
<th>Description</th>
<th>Extensions</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>.cdlms, .CBLIMS</td>
</tr>
<tr>
<td></td>
<td>IMS/TP COBOL Delta Macro Programs</td>
<td>.cdlms, .CDMIMS</td>
</tr>
<tr>
<td></td>
<td>IMS/TP PL/I Programs</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 are missing in the Enterprise View Knowledge Database, you can create the same or other extensions using the Work with File Extensions menu in the Environment Settings part of Enterprise View Configuration Manager. This operation can be performed only by an Administrator of the product.

### 4.4. Missing Sources

If sources are not provided, they cannot be analyzed. When the source is missing and referenced by any other source, an unresolved object is created. Many other issues can be generated as well. A set of the most important problems is listed below.

#### 4.4.1. Missing Programs or Transactions

**Issues:**

- An unresolved object of a 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.

#### 4.4.2. Missing Copy/Include Books

**Issues:**

- If referenced, an unresolved object of a specific type will be generated.
- Metrics without copy/include books will be corrupted.
4.4.3. Missing JCLs and JCL Catalogued Procedures

Issues:
- Since the JCLs are starting points, the chain between 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.

4.4.4. Missing IMS Stage One

Issues:
- The transactions and programs which are declared will be lost.
- The relationships between transactions and programs which are declared will be lost.
5. Defining an Organization View Tree

The organization view is an Enterprise View structure. It is used to better represent the structure of the sources. You can organize the applications in technical or custom organization views. The technical organization view is used for:

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

You can create a custom organization view. This organization view enables you to see the analysis results from a different point of view, reordering and connecting the components from one or some technical organization views in one custom view. You can combine the components according to criteria, different from the developer’s point of view and closer to his/her business environment.

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

Bear the following in mind when an organization view is defined for sources running in an IMS environment on a mainframe platform:

- System OS/390 has to be selected when a technical organization view is defined.
- A custom organization view can be created at any time of 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 the Organization View Tree. They are described in chapters dedicated to each host job. The newest version of the Enterprise View product has a Job Creation Wizard for entering the starting parameters of the host jobs. The Job Creation Wizard is started from the Work with Jobs icon or from the Organization View Tree. The sequence of the Job Creation Wizard menus is presented below:

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

![Job Wizard Welcome menu](image)

2. Job Wizard selection menu that contains all types of jobs. You can select the name of the host job by choosing from the dropdown list. The schedule date and time can be set too. You can click Next to continue or Back to review the previous choice. If you click Cancel, the Job Wizard will terminate. The host jobs that are the object of examination in this document are:

   - Inventory probe Job
   - Inventory Job
   - Analysis Job
   - Life Cycle Job
   - Inventory Deletion Job
   - HTML Export Job
   - End Daemon Job
3. Job Wizard menu for parameter selection. You can select the package, the organization view, the operational mode and the current period from the dropdown 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 the previous choice. If you click Cancel, the Job Wizard will terminate.

4. Job Wizard menu for selecting a Multidaemon which will execute the submitted job, and entering the parameters of the submitted job. There are two Daemon jobs: MAIN and SECONDARY. You can click Generate to continue or Back to review the previous choice. If you click Cancel, the Job Wizard will terminate.
5. Job Wizard complete menu. Click Finish to exit the Wizard.

If you have submitted the job for execution using the organization view tree, it will start after you click Finish. If the job has been submitted from the Work with Jobs icon, the Work with Jobs menu will appear.

Depending on the Daemon name selected in the Wizard menu, you must start the appropriate Multidaemon job and click OK or Apply to submit the job.
7. Starting the Inventory Probe Job

7.1. Introduction

The Inventory Probe Job verifies if all sources provided to Micro Focus Enterprise View by means of the Define Technical View function have an extension recognized by the Micro Focus Enterprise View Knowledge Base.

As described in the chapter Providing Sources to Micro Focus Enterprise View Server Machine, the product needs the sources which are to be processed to be put 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 there in a package?
- How many sources are there for every type of source?
- What types of sources are there in the package?
- Are there unrecognized and unsupported extensions (i.e. types of sources)?

The result of the Inventory Probe is a report that contains the number of sources in the package, sorted by extension, highlighting which are the supported and unsupported 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 the Inventory Probe Job.

7.2. How to Run the Inventory Probe Job

There are two ways of running the Inventory Probe job. In both cases, the Job Wizard will start if you work with the newer versions of the Micro Focus Enterprise View.

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 popup window and click OK. The Work with Jobs menu can be selected using Enterprise View Configuration Manager, Enterprise View Builder and Enterprise View View Tech solutions.
The window with fields which are not populated 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 populated 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 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 information about the selected package.

- Start a Multidaemon session on the server machine.
- Click OK from the Work with Jobs window to start the job.

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

7.3. Understanding Inventory Probe Job Results

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

- **Catalogued**

  All Objects belonging to this category will be inventoried. Depending on the Analysis and Save mode, different actions will be taken:

  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, detailed analysis only for the supported language. For PL/I sources, detailed analysis 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. It is also not possible to display their source code by means of the Show Source function or to trace source history because it has 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 will be ignored. This group contains sources recognized in the Enterprise View Tables but not included in the Inventory nor in the Repository.  

b) Undefined Files - Sources corresponding to the extensions described in this group do not exist inside the Enterprise View configuration tables, therefore they will not be included in the inventory, nor analyzed.
The file gives a broad view of what is present in the package and helps to check immediately if all the components necessary to process the user’s environment have been included in the package.

**Note:** We advise 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.

### 7.4. When to Rerun the Inventory Probe Job

We advise you rerun the Inventory Probe Job until there are no more wrong or undefined extensions and missing sources.

#### 7.4.1. Rerun after Adding New Components

If any sources have been omitted and are not present in the package, it is important to add them by means of 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. By executing Inventory Deletion Job**

The following steps have to be performed:

1. Run **Inventory Deletion Job with checking the Delete Structure checkbox** 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 be transferred to the corresponding existing folders too.
4. Define technical organization view including the new folders as components.
5. Run the Inventory Probe Job again and check the results.

**II. Without executing Inventory Deletion Job (for advanced users)**

The following steps have to be performed:

1. Create a new folder for the missing sources of the application on the server machine. If there are sources with different types of file extensions, the folder can contain subdirectories.
2. Transfer the missing sources to the corresponding new folders and check the file extensions.
3. Define a 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 (package).

Run the Inventory Probe Job again and check the results.

#### 7.4.2. Rerun after Adding New Extensions

The sources catalogued in Undefined Files MUST BE carefully considered because their extensions are not recognized in the Enterprise View Knowledge Base. If these sources are necessary for the environment, you can add their extensions in the Enterprise View File Extension Table (for details, refer to Work with File Extensions in the Enterprise View Configuration Manager 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 an incorrect assignment of the extensions to sources, you must correct the wrong assignment extensions and rerun the Inventory Probe Job.

**Note:** We advise you rerun the Inventory Probe Job after adding new extensions in the Work with File Extensions table to check that the Enterprise View product has properly understood them.
8. Starting the Inventory Job

8.1. Introduction

The Inventory Job loads information about all the sources contained in your folders by means of 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 in 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, indexes, 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.

You must perform an inventory on a package before running the subsequent phases. The Inventory Job can be rerun if necessary. For more information, see the chapter *When to Rerun the Inventory Job*.

8.2. How to Run the Inventory Job

There are two ways of running the Inventory Job. In both cases the Job Wizard menu will start if you work with the newer versions of the Enterprise View.

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 Job from the popup 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 the selected package, choose the Job Submission item and select Submit Inventory Probe Job 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 information about the selected package.

- Start a Multidaemon session on the server machine.
- Click OK in the Work with Jobs menu to submit the job.
8.3. Inventory Job Results

The results after completing the Inventory Job can be viewed in 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 every row of the specific tabsheet for each source class and using the dropdown menu.

In the next picture, the Summary tabsheet of Enterprise View Builder is displayed.

There is detailed information about all loaded objects in the All Objects tabsheet of Enterprise View Builder, as illustrated in the next picture.
The information written for every object in the Enterprise View Tables is:

- about the qualification of the identification object:
  - name, alias and description
  - Enterprise View system, class, type, and language

- about the metrics:
  - number of the objects
  - source lines of code (SLOC)

- about 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 types 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 ‘not catalogued’ category in the Inventory Probe Job results. These sources exist in 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 every object after the 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 after 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)** - execution of the Inventory Job has been successfully completed.
- **Disabled** - inventory analysis has not been executed. The object has been recorded in the Enterprise View Repository as an object for reference only.

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

### 8.4. When to Rerun the Inventory Job

We advise you rerun the Inventory Job in the following cases:

1. **Errors can occur 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 checking the Delete Structure checkbox** to delete all the information from the Enterprise View Repository and Database tables except the structure of the selected organization view.
2. Correct the errors.
3. Run the 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 the Inventory Job. In this case, all problems regarding wrong or undefined extensions and missing sources will appear after executing the Inventory Job.

If there are missing sources, you can follow the instructions described in the chapter [Rerun after Adding New Components](#) and rerun the Inventory Job instead of the Inventory Probe Job.

In case of incorrectly assigned extensions to sources or missing extensions, follow these steps:

1. Run **Inventory Deletion Job without checking the Delete Structure checkbox** to delete all the information from the Enterprise View Repository and Database tables except the structure of the selected organization view.
2. Correct the wrong extension or add missing extensions in the Enterprise View Knowledge Database.
3. Run Inventory Probe Job (advisable).
4. Run Inventory Job again and check the results.
9. Starting the 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 on the basis of 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 the Work with Codes function of Environment Settings. For more information, see the Enterprise View Configuration Manager reference guide.

During analysis, Enterprise View:
- identifies the definitions of the areas with the corresponding fields when it is possible.
- identifies the relationships between objects.
- identifies the dynamic links between the objects if there are dynamic calls in the source codes.
- 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 your environment (IMS Batch or IMS Online) have been included in the same package.

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

Otherwise you must run the analysis following 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.

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

- Enterprise View Tech
- Enterprise View Builder
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 if necessary. For more information, see the chapter When to Rerun the Analysis Job.

9.2. How to Run the Analysis Job

There are two ways of running the Analysis Job. In both cases the Job Wizard menu will start if you use the newer versions of Enterprise View.

1. Using the Work with Jobs menu:
   - Click 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 Analysis from the popup window and click OK.

   - The window with fields, which are not populated, 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 populated fields is ready for execution.

The Organization View box allows you to select a package, application, application version or component for analysis.

| Model ID: | Test IMS/DLI Applications |
| Package:  | IMS_Application |
| Organization View: | IMS Stage1 |
| Operational Mode | Includes |
| Update Trend | IMS Stage1 |
|             | Includes |
|             | IMS Stage1 |
|             | FMTs |
|             | DDL |

2. Using the Organization View Tree:

- Select a package, application, application version or component in the organization view tree.
- Right-click the selected package, application, application version or component, choose the Job Submission item and select Submit Analysis Job 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 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 objects’ metrics.
- Start a Multidaemon session on the server machine.
Click OK 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 if everything has been processed properly, if there are errors found, or if there are problems that require your intervention. The Analysis Status field in Enterprise View Builder and Enterprise View Tech shows the results after the 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 - the analysis is not 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 an 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 that have 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, the following considerations have to be taken 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, time out and not parsed) have to be considered carefully.

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 presented below.

The All Objects tabsheet shows the Analysis Status of all objects in the selected organization view. You can see the source code of the selected object by using the Show Source option from the popup 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, the Enterprise View product 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, CICS resource, SQL or DL/I database table. Some cases of missing objects are described in the examples below.

**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 the Enterprise View product 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, for example, JCL file definition, CSD resource definition file, or DB2 DDL table definition, has been included in the package.

When the Enterprise View product 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 the Enterprise View product 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 the Enterprise View product.
- The Enterprise View product 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.
- The Enterprise View product will not create relationships between these objects.

**Note:** The list of unresolved objects is very important as documentation for the missing objects.

Usually, your applications contain thousands of objects and sometimes it is not easy to identify immediately 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 of the sources in the development environment

**9.3.1.1. How to Manage Unresolved Objects**

The most common causes of unresolved objects and the ways of managing them are described in this chapter.

**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 part to the package using the Define Technical View function and rerun the Analysis Job following the necessary steps as explained in the chapter [When to Rerun the 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 necessary steps described in the chapter When to Rerun 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 filling in the LOADERS section of the configuration file JclConfig.cfg with the names of all loaders that the application uses as shown in the next example. The JclConfig.cfg file is in the JOB directory on the Enterprise View server machine.

The LOADERS section of the file JclConfig.cfg:

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

[LOADER]
TYPE = IMS
NAME = 'DFSRRC00'

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

[LOADER]
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, you can find out if the source has been included in the organization view or not.
- Using the Browse Object function from the dropdown menu, you can follow the reference and display the line where the item was involved.
- Using the Reference section you can see all the characteristics of this reference and the part from which the unresolved object originates.

In the next picture the Show Source function shows the source of the object (in this case JCL file) and the line at which the unresolved card’s object is called.

![Image of Enterprise View Tech interface showing unresolved objects and their details.]
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 the Unknown Syntax Job. This job reports the names of all objects that contain unsupported or unrecognized syntax. For more information see the chapter Starting the Unknown Syntax Job.
2. Select the Unsupported Syntax tabsheet in Enterprise View Builder and verify for each object the type of syntax not understood by the Enterprise View product.
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 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 (time out)

This error appears after the execution of Analysis Job and means that the analysis of the object has not been completed because there has been no answer for a period longer than 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 the 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 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 may remain with status To Do (Extended) after the execution of an analysis job. Possible reasons for this can be:

I. Messages for:
   - Wrong extension for the type of source.
   - Unsupported syntax lines in the source.

   In this case the following 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 checking the Delete Structure checkbox to delete all objects with the 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 Analysis Job for the package.

II. Analysis Job does not run for some components

   If the analysis is not executed at package level, some components are skipped by the analysis. In this case run the Analysis Job for the components never analyzed taking into 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 following the instructions in the chapter When to Rerun the Analysis Job.

III. Analysis Job has stopped for some reason

   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 has to be rerun in the following cases:

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 (time out)
   - Sources with analysis status Done (not parsed)
   - Sources analyzed in a wrong order
   - Sources with unknown syntax
   - Other errors (not enough space for writing, database problems or Analysis Job has stopped for some reason)
The following steps have to be performed:

1. Run **Inventory Deletion Job without checking the Delete Structure checkbox** to delete all objects with the detailed information about them from the Enterprise View Repository and Database tables for the selected package except 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 a package level, keep the right order for analysis of components.

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

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

- **By executing Inventory Deletion Job with checking the Delete Structure checkbox**
  
  The following steps have to be performed:
  
  1. Run **Inventory Deletion Job with checking the Delete Structure checkbox** to delete all objects with the detailed information about them from the Enterprise View Repository and Database tables for the selected package together with the structure of the organization view.
  2. Transfer the missing sources in the new or existing folders and check the file extensions.
  3. Define 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 checking the Delete Structure checkbox**
  
  The following steps have to be performed:
  
  1. Run **Inventory Deletion Job without checking the Delete Structure checkbox** to delete all objects with the detailed information about them from the Enterprise View Repository and Database tables for the selected package except the structure of the organization view.
  2. Transfer the missing sources in the existing folders only 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.

**Note:** 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 process: inventory and analysis phases for the selected organization view. It can be repeated directly in cases of errors, unresolved objects (missing sources), or wrong extensions. This job is executed at package level only. Life Cycle Job tracks if there are changes in the application (for example new sources, deleted sources or sources with changed program code) and updates the information in the Enterprise View Repository and Database tables. This job performs inventory and analysis functions:

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

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

10.2. How to Run the Life Cycle Job

There are two ways of running the Life Cycle Job:

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 Life Cycle from the dropdown window and press the OK button.
2. Using the organization view tree:

- Select the package in the organization view tree, right-click and choose the Job Submission item and select Submit Life Cycle Job. 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 Multidaemon session on the server machine.
- Click OK on the Work with Jobs window to submit the job.
10.3. Understanding the Life Cycle Status Results

The results of the Life Cycle Job can be viewed in Enterprise View Builder and Enterprise View Tech. The Status field in the tabsheets for every class of objects (programs, copy/includes, JCLs, transactions, maps, etc) shows the changes after the 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 an 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 has to be started after execution of the Analysis Job. It works at any 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 a 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 Enterprise View Configuration Manager. 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 Unsupported Syntax Job

This job can be run in the same way as the Life Cycle Job. But instead of the Life Cycle Job, the Unknown Syntax Job has to be submitted.

11.3. Understanding Unsupported Syntax Results

A list of all objects that contain unknown syntax lines can be viewed using the Unsupported Syntax tabsheet of Enterprise View Builder. 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 recognized as unknown syntax.

The number of the line with unrecognized syntax for every object will appear by clicking on the object containing such syntax in Enterprise View Tech.

How to solve the problem:

The possible reasons can be:

- Wrong extensions for the type of source: Correct the extensions and rerun the job.
- Wrong syntax in the source code: Correct the syntax and rerun the job.
- Binary or other unspecified text in the source: Exclude the object with such 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 any 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 except the structure of the organization view, when the Delete Structure checkbox is not checked.
- When 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, whereupon it deletes the structure of the selected level of the organization view: package, application, application version or component.

12.2. How to Run Inventory Deletion Job

This job can be run using an organization view tree. In the newer Enterprise View versions this job can be started from the Work with Jobs menu too.

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. If the Job Creation Wizard menu appears, go to the chapter Starting Host Jobs. Otherwise the job will appear in Work With Jobs where you can edit its parameters.

Check the Delete Structure checkbox, if it is necessary. Start a Multidaemon session on the server machine. Click OK or Apply to submit the job.

**WARNING:** Execution of the Inventory Deletion Job is not recommended during the second phase of the analysis process (Life Cycle process) when the Trend box is checked during the analysis, because the trend information (previous status and the movement of the objects’ metrics) will be lost.
13. Starting the HTML Export Job

This job is used for exporting organization view structures with some statistics in HTML format. HTML export can be performed at any level of the technical organization view: package, application, application version or component. The job can be started by using the Work with Jobs menu or the Documentation Export of the Solutions part of Enterprise View Navigator. By checking the Refresh Existing HTML Export checkbox, the user can see for which organization views or their levels in the package, HTML documentation export already exists. If the user starts a job for a package that already has exported documentation, only the information about updated objects will be added to the existing exported documentation.

Three general types of information are reported:
- Overview information
- Object list information
- Detailed information about objects.

Starting index.html file, the user can 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 stop the execution of the started Multidaemon. This job performs the following operations:

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

This job can be started using the Job Creation Wizard menus only.
15. Analyzing IMS Sources

15.1. IMS General Considerations

IMS (Information Management System) is a hierarchical database management system running on mainframe systems z/OS, OS/390, MVS. IMS contains two major components that can be used together or separately: the IMS DB and the IMS TM. They both can be used together as DB/DC environment.

IMS DB is the Database Manager system, which processes concurrent database calls.

IMS TM (or IMS DC - data communication system) is the Transaction Manager system. It provides transaction processing for both IMS DB and DB2 UDB for z/OS databases.

15.2. Sources Containing File and Database Definitions

15.2.1. Introduction

The following languages are used for describing the definitions of system files and databases:

1. Data Definition Language (DDL) for DB2 Database.
2. Database Description Language (DBD) for DL/I Database.
3. VSAM Delete/Define Language for VSAM files.

During the analysis of the IMS sources containing DB2, DL/I, or VSAM Delete/Define definitions, Enterprise View scans the sources and loads in the Enterprise View Repository information about the files and database definitions as well as about the following measures:

- Lines of Code - LOC
- Source Lines of Code - SLOC
- Comment Lines of Code - CLOC
- Change Traffic

**NOTE:** The analysis of the database definitions must be executed as a first step of the analysis (when the analysis is not at package level). Otherwise the Enterprise View product creates 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 Database

DL/I is a hierarchical database working on MVS platform. Database Description Language (DBD) is used for describing the components of the DL/I database - physical files, segments, program specification blocks (PSB), and program communication blocks (PCB).

The sources containing DBD sources must have an extension .dbd or .DBD to be properly analyzed by the Enterprise View parser. The analysis type of the DBD sources is set as ‘Extended’ in Enterprise View Knowledge Database.
During the DBD analysis the Enterprise View parser creates:

- The DBD list
- The SEGMENTS list and their hierarchy.

For more information about the DL/I database elements and calls in IMS and CICS environment see DL/I Process Guide.

### 15.2.3. DB2 Database

DB2 is an enhanced relational database working on Mainframe and Windows platforms. The Data Definition Language is used for describing the components of the DB2 database (tables, views, indexes, alias, etc).

The sources containing DDL statements must have an extension `.ddldb2` or `.DDLDB2` to be properly analyzed by the parser. The analysis type of the DDL sources is set as ‘Extended’ in Enterprise View Knowledge Database.

The Enterprise View parser extracts and loads in the Enterprise View Repository the database definitions defined in DDL sources - tables, indexes, views. Where it is possible, a list of fields of each file is loaded.

The following database components are described using data definition language (DDL):

- Create Table
- Create View
- Create Index
- Create Alias
- Create Synonym

### 15.2.4. VSAM Delete/Define File Statements

A VSAM file is known as a cluster. VSAM clusters are managed by a utility called IDCAMS. VSAM stands for Virtual Storage Access Method and allows sequential or direct access to the data in the files.

IDCAMS utility control statements are used for describing the VSAM clusters. The file containing the source code of the VSAM Delete/Define File statements must have an extension `.ddef` or `.DDEF` to be properly analysed by the Enterprise View Parsers. Usually VSAM statements are located in JCL SYSIN DD * statement or are written as member (card file) of the partitioned data set that is included in input DD statement during the execution of the IDCAMS step.

The Enterprise View parser extracts and loads in the Enterprise View Repository VSAM Cluster and Path definitions as a physical file and VSAM Alternate Index definitions as a logical file.

### 15.3. Copy/Include Books

The copy/include books are external sources containing program code. The COPY INCLUDE statements or directives are used in IMS programs to incorporate the external source text into the source programs at specific points. The include books can contain:

- Assembler, C/C++, COBOL or PL/I program code or declarations
- SQL statements
To be properly analyzed by the Enterprise View parsers, the copy/include books must have the following extensions:

- For Assembler language: .asminc or .ASMINC
- For C language: .h or .H
- For C++ language: .hpp or .HPP
- For COBOL language: .cpy or .CPY
- For PL/I language: .inc or .INC

The analysis of the include books is not a prerequisite for the analysis of the programs because the program analysis recognizes the pre-processing of all include books called by the programs. Therefore if an unresolved object is an include book, this means that the include book is missing or that it has been loaded with a wrong extension.

In the client modules (i.e. Enterprise View Tech) all relationships retrieved in the program analysis are displayed as belonging to the program. The statement number indicated will be the original statement where the copy/include book is included.

The analysis type of the copy/include sources is set as ‘Basic’ in Enterprise View Knowledge Database for Assembler, C/C++, COBOL and PL/I languages. The Enterprise View parser scans copy/include books and it loads in the Enterprise View Repository the following measures:

- Lines of Code - LOC
- Source Lines of Code - SLOC
- Comment Lines of Code - CLOC
- Change Traffic

The status of the copy/include files after the inventory and analysis phase can be viewed using the Copy/Includes tabsheet of Enterprise View Builder or Enterprise View Tech.

### 15.4. Sources with SQL Statements

During the analysis, when the Enterprise View parser meets some SQL statements, it starts the SQL analyzer which retrieves the types of relations between the programs and the objects found in the SQL statements. Such analysis is performed for COBOL and PL/I sources only in IMS environment.

The statement number that indicates the line of source that generates a relationship (i.e. program-file relationship) is always the statement number in which the EXEC SQL statement begins in the original source. For SQL statements present in copy/include books, the reported statement number is the number of the including in the original source copy/include book.
16. IMS Batch Environment

16.1. Introduction

Batch processing is the execution of a series of programs (jobs) on a computer without human intervention. There are two batch environments in IMS. They are:
- Database manager batch - DBB
- Transaction manager batch - TMB

The batch environment consists of a batch region where an application program and DL/I routines reside. The batch jobs that run in this environment are initiated with JCL.

Job Control Language (JCL) is very important for the batch processing. It is used as the starting point for the call to the programs and procedures. Enterprise View determines not only the relationships between a batched job and programs executed in the steps but also the link between the physical name (DSNAME) and the logical name (DDNAME) of the files, used by the programs in the JCL chain.

The analysis of the JCL is therefore a prerequisite for the analysis of the programs.

16.2. Batch Jobs (JCL) and Catalogued Procedures

16.2.1. Introduction

During the analysis of the sources containing JCL statements, preprocessing of the JCL is performed to gather all the information necessary to the Enterprise View Parser.

The information collected in this phase regards to:
- Catalogued procedures
- Cards
- Path list of the catalogued procedures
- Path list of cards
- Loaders

Catalogued procedures contain JCL statements. They should be available in the package because they are expanded in the JCL sources, otherwise Enterprise View will load them as unresolved objects.

Card files are names of partitioned data set (PDS or extended PDSE) members. They should be available in the package because they are found in JCL DD control statements, otherwise Enterprise View will load them as unresolved objects.

The loaders are system utilities or database interface programs whose source is never available. For this reason they cannot be available in the package and usually Enterprise View loads them as unresolved objects and the direct link between the batch jobs and the program called by the loader will be lost. To avoid this problem the user can describe the loader names in a special configuration file as shown below.
The path list of catalogued procedures and cards as well as the loader names can be described in a special configuration file named `JclConfig.cfg`. This file is situated in the Job directory on the Enterprise View server machine. The user must provide the last three pieces of information.

The configuration file is divided in 3 main sections:

- **INCLUDE PATHLIST** – This section specifies the path list to be used as a search list for inclusions. It contains the name of the directory where catalogued procedures are situated on the Enterprise View server machine.

- **CARD PATHLIST** – This section specifies the path list (the name of the directory on the Enterprise View server machine) that will be used as a search list for cards.

- **LOADER LIST** – This section contains a list of all loaders used in the organization view.

The configuration file `JclConfig.cfg` contains the names of some common loaders such as IKJEFT01 or IDCAMS. The user should verify (before proceeding with the analysis phase) if the environment uses some of these loaders or if any other loaders are required.

There are different classes of loaders:

- **DB2**
- **IMS**
- **IMSDB2**
- **IDC**
- **SYSTEM UTILITIES**

The configuration file `JclConfig.cfg` (preloaded with the product) contains the names of some common loaders such as IKJEFT01 or IDCAMS. The user should verify (before proceeding with the analysis phase) if the environment uses some of these loaders or if any other loaders are required.
synonyms of standard loaders. In case that there are such programs, the user must add them to the configuration file.

Enterprise View Parser scans the batch job sources and the catalogued procedures and loads in the Enterprise View Repository the following main information:

- Physical file (data set) names in the batch job (JCL)
- Relationships between JCL and Catalogued Procedures
- Relationships between JCL and Cards
- Relationships between JCL and Programs
- Relationships between called programs and files

16.2.2. JCL Physical Files

Data sets that belong to DL/I databases (or an IMS Monitor data set) are specified with JCL DD statements.

If during the analysis of the JCL sources, Enterprise View meets descriptions of the data sets, these files will be recorded as physical files in the Enterprise View Repository. The relations between these files and the programs that use them are created on the basis of the correspondence between DSNAME/DDNAME names in DD statement of job control language and the file description name in the program.

The analysis type of the JCL files (Sequential, Temporary, GSAM) is set as ‘Disabled’ in Enterprise View Knowledge Database. Therefore they are not analyzed. The file names are included in the Repository for inventory purposes only.

Enterprise View gives information about JCL files (if there are any) after the execution of the analysis. The status of the physical files after the analysis phase can be viewed using the Files tabsheet of Enterprise View Builder or Enterprise View Tech

16.2.3. GDG (Generation Data Group) File Definitions

A generation data set is one of a collection of successive, historically related, cataloged data sets. They are known as a generation data group (GDG). When a generation data set is created, the relative generation number tells the system whether this is the first data set being added during the job, or the second, or the third, etc.

In this example a description of the GDG data set with generation number +2 is shown in JCL DD statement.

```//DFSUDUMP DD DSN=B072.BACK01.DIARSUBQ(+2),DISP=OLD```

When, during the JCL analysis, Enterprise View finds a file where the version is specified, this file will be stored as GDG.

If two different batch jobs refer to the same GDG file but in the first one the version is specified, while in the second – the version is not specified, the first JCL will be loaded as a GDG file, while the second will be loaded as a sequential file.

The analysis type of the GDG files is set as ‘Disabled’ in Enterprise View Knowledge Database. Therefore they are not analyzed. The GDG file names are included in the Repository for inventory purposes only.

The status of the GDG files after the inventory or analysis phase can be viewed using the Files tabsheet of Enterprise View Builder or Enterprise View Tech.
16.2.4. Catalogued Procedures

Procedures are JCL skeletons and provide macro ability to JCL. They are used for executing the same set of job control statements repeatedly with little or no change. A cataloged procedure is a named set of job control statements.

The catalogued procedures must have an extension `.proc` or `.PROC` to be properly handled by Enterprise View.

The analysis of the catalogued procedures is not a prerequisite for the analysis of the JCL sources, because the Enterprise View product foresees the pre-processing of all procedures included by the JCL. If catalogued procedures appear as unresolved objects, this means that these procedures are missing or that they have been loaded with the wrong extensions.

The analysis type of the catalogued procedures is set as ‘Extended’ in Enterprise View Knowledge Database. The Enterprise View parser scans the catalogued procedures and loads in the Enterprise View Repository the following measures:

- Lines of Code - LOC
- Source Lines of Code - SLOC
- Comment Lines of Code - CLOC
- Change Traffic

The status of the catalogued procedures after the inventory or analysis phase can be viewed using the JCLs tabsheet of Enterprise View Builder or Enterprise View Tech.

At the end of the analysis phase the following information is gathered:

- JCLs – Catalogued Procedures relationships.

This information is available in the References and Browse Object functions of Enterprise View Tech. It is important to point out that by means of this function it is possible to identify the line in the source code where the relationship is generated.

16.2.5. JCL Cards

A card, according to Enterprise View Knowledge Base extensions’ classification, is the name of a member of partitioned data set (PDS or PDSE). The member name is recorded in parenthesis after the name of the partitioned data set in DSNAME parameter of the JCL DD statement. The member can contain program code, text or input statements for SYSIN DD statement.

For example:

```bash
//SYSIN DD DSN=Z005.SYSINLIB(B072RE2Q),DISP=SHR
```

An external file named `B072RE2Q` has to exist in the sources. It should have an extension `.card` or `.CARD` to be properly handled by Enterprise View.

The analysis of the card files is not a prerequisite for the analysis of the JCL sources because the Enterprise View product foresees the pre-processing of all the cards included by the JCL.

Therefore, if unresolved objects are card files, this means that they are missing or that they have been loaded with the wrong extensions.
The analysis type of the JCL Card files is set as ‘Basic’ in Enterprise View Knowledge Database. The Enterprise View parser scans card files and loads in the Enterprise View Repository the following measures:

- Lines of Code - LOC
- Source Lines of Code - SLOC
- Comment Lines of Code - CLOC
- Change Traffic

The status of the JCL sources containing card files after the inventory or analysis phase can be viewed using the JCLs tabsheet of Enterprise View Builder or Enterprise View Tech.

16.2.6. JCLs

JCL is a low-level language source with instructions known as job control statements. The job control statements are interpreted, not compiled. MVS users use JCL to define work requests (called jobs) and to submit these jobs to the system.

When there are IMS calls in some of the JCL steps, the parameters passed to the loader in the EXEC PGM statement are:

**Case 1**

```
// EXEC PGM=DFSRRC00, PARM=('Par-Dli,Pgm,Psb') or
// EXEC PGM=DFSRRC00, PGM=Pgm, PSB=Psb
```

Where: Par-Dli can be one of the following: DLI, UDR, ULU, BMP, DBB

The running program is DFSRRC00 and only the parameters, in one of the described formats must be interpreted. In this case, the additional information that Enterprise View will extract is:

- Program - indirectly called

**Case 2**

In this case an invocation to an IMSDLI loader is made:

```
//STEP02 EXEC PGM=DFSRRC00,
//       PARM=(DLI,DSNMTV01,PSBNAME,,0000,,0,,
//DFSRESLB DD DSN=IMS.ITIMS1LO.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DDITV02 DD *
P1,P2,P3,P4,P5,P6,P7,P8,PGMNAME
```

The running program is DSNMTV01. The program actually invoked is the 9th parameter in the DDITV02 DD statement. In this case, the information that Enterprise View will extract is:

- Program DSNMTV01 - indirectly called
PGMNAME – indirectly called

The names DFSRRC00 and DSNMTV01 can have synonyms; the corresponding rows in loader list file are made up of pairs of names.

The JCL sources must have an extension .jcl or .JCL to be properly analyzed by the Enterprise View parsers.

In the client modules (i.e. Enterprise View Tech) all the relationships retrieved in the JCL analysis are displayed as related to the JCL source, even if the statement that gives the relationship originally belongs to a catalogued procedure or a card.

If a relationship originates from a statement that belongs to a procedure or a card, the related statement number shown in the Enterprise View client solutions will be the JCL statement where the card or procedure has been included.

The analysis type of the JCLs is set as ‘Extended’ in Enterprise View Knowledge Database. The Enterprise View parser scans the JCLs and loads in the Enterprise View Repository tables the following measures:

- Lines of Code - LOC
- Source Lines of Code - SLOC
- Comment Lines of Code - CLOC
- Change Traffic

The status of the JCL sources after the inventory or analysis phase can be viewed using the JCLs tabsheet of Enterprise View Builder or Enterprise View Tech.

At the end of the analysis phase the following information is gathered:

- JCLs – Catalogued Procedures relationships
- JCLs - Programs relationships

This information is available in the References and Browse Object functions of Enterprise View Tech. It is important to point out that by means of this function it is possible to identify the line in the source code where the relationship is generated.

16.2.7. Batch Programs

Batch programs become active when a calling program invokes the main procedure. This calling program usually is the operating system, but it could be another program.

Enterprise View analyses the sources written in Assembler, C/C++, COBOL, Cobol Delta Macro and PL/I in batch IMS environment.

To be properly analyzed by Enterprise View Parsers the programs running in IMS batch environment must have the following extensions:

- For Assembler language: .asmims or .ASMIMS
- For C language: .cims or .CIMS
- For C++ language: .hppims or .HPPIMS
- For COBOL language: .cbl, .cblims or .CBL, .CBLIMS
- For Cobol Delta Macro language: .cdmims or .CDMIMS
- For PL/I language: .pli, .pliims or .PLI, .PLIIMS

The analysis type in Enterprise View Knowledge Database for the different programming languages working in IMS/Batch environment can be basic or
extended and will be examined in the chapter Enterprise View Analysis Types for IMS Programs.
17. IMS Online Environment

17.1. Introduction

There are three online environments in the IMS system:

- DB/DC (Database/Data communication) environment – an IMS Database Manager and Transaction Manager are used together.
- DCCTL (Data Communication Control) environment - an IMS Transaction Manager subsystem that has no database components. It does not service DL/I database calls.
- DBCTL (Database Control) environment – an IMS Database Manager with transaction management subsystem (for example, CICS). DL/I region owns the databases to be processed.

The resources that Enterprise View analyses in this environment are:

- IMS Stage One file - it contains the definitions of the resources of the IMS environment
- IMS/TP Programs
- IMS/TP Transactions

17.2. IMS Resource Definitions

The IMS Stage One file must have an extension .ims or .IMS to be properly analysed by the Enterprise View Parsers. The analysis type of the IMS Stage One is set as ‘Extended’ in Enterprise View Knowledge Database. Enterprise View parser scans the IMS Stage One file and loads in the Enterprise View Repository the following measures:

- Lines of Code - LOC
- Source Lines of Code - SLOC
- Comment Lines of Code - CLOC
- Change Traffic

This information is available in Enterprise View Builder and Enterprise View Tech and can be viewed by using the Others tabsheet.

At the end of the analysis the following information can be viewed:

- Transaction names
- Related program names
- The type of the relation

The information can be viewed in Enterprise View Tech client solution using Programs, Transactions and Others tabsheets. By means of the References and Browse Object functions, it is possible to identify in the source code the line that generates the relationship.

This information is extracted from an APPLCTN and TRANSACT macro of IMS Stage One file, when the keyword-parameter PGMTYPE=(TP) is found in APPLCTN
macro. When this parameter is: PGMTYPE=(BATCH), the program names are skipped.

For example:

*       SQ03     REGULAR PAYMENTS FP
APPLCTN
PSB=KSQF02M0,PGMTYPE=(TP),SCHDTYP=PARALLEL,FPATH=NO
TRANSACT CODE=TSQ00300,PRTY=(4,8,10),INQ=(NO,RECOVER),
FPATH=NO,PARLIM=1,PROCLIM=(5,5),MODE=SNGL,EDIT=(UC),
SCHD=1,SEGSIZE=2048,SEGNO=512,MAXRGN=3,
MSGTYPE=(MULTSEG,RESPONSE,10)

*       SQ02     REGULAR PAYMENTS BMP
APPLCTN PSB=KSQ26PI0,PGMTYPE=(BATCH),SCHDTYP=PARALLEL

In this example the program name KSQ26PI0 will be skipped by the Enterprise View parser.

If the analysis finds program sources not included in the package, they are loaded as unresolved objects.

17.3. IMS MFS Definitions

In IMS, screen formats are stored in IMS library. They are called into the MFS buffer pool for online execution when it is necessary.

The MFS must have an extension: .mfs, .mapmfs or .MFS, .MAPMFS to be properly recognized and handled by Enterprise View Parser. There is no specific parser for these objects.

If during the analysis of IMS programs the Enterprise View product does not find MFS definitions called by the programs, these map definitions are stored as unresolved objects.

The analysis type of the IMS MFS definitions is set as ‘Disabled’ in Enterprise View Knowledge Database. Therefore they are not analyzed. The map names are included in the Repository for inventory purposes only. The Object Change Traffic information (New/Changed/Deleted) is recorded too.

The status of the MFS sources after the inventory or analysis phase can be viewed using the Maps tabsheet of Enterprise View Builder or Enterprise View Tech.

17.4. IMS/TP Programs

The programming languages that Enterprise View supports for analysis in online IMS environment are: Assembler, C/C++, COBOL, Cobol Delta Macro and PL/I.

To be properly analysed by Enterprise View Parsers the programs running in IMS/TP environment must have the following extensions:

- For Assembler language: .asmims or .ASMIMS
- For C language: .cims or .CIMS
- For C++ language: .hppims or .HPPIMS
- For COBOL language: .cblims or .CBLIMS
For COBOL Delta Macro language: `.cdmims` or `.CDMIMS`

For PL/I language: `.pliims` or `.PLIIMS`

The analysis type in Enterprise View Knowledge Database for the different programming languages working in IMS/TP environment can be basic or extended and will be examined in the chapter Enterprise View Analysis Types for IMS Programs.

### 17.5. IMS Transactions

The transaction is the basic unit of work in an IMS online environment. Transaction processing consists of:

- Receiving a request for work that has been entered at a terminal.
- Invoking a program to do the work, and preparing a response.
- Transmitting the response to the terminal that requested the work.

The simplest kind of transaction involves two messages: an input message from the terminal user and an output message in return. Application programs can also send messages to terminals other than the input source, and they can generate transactions.

The transactions running in IMS environment must have an extension `.trn` or `.TRN` to be properly analysed by Enterprise View Parsers.

The analysis type of the IMS transactions is set as ‘Disabled’ in Enterprise View Knowledge Database. Therefore they are not analyzed. The transaction names are included in the Repository for inventory purposes only. The Object Change Traffic information (New/Changed/Deleted) is recorded too.

The status of the transactions after the inventory or analysis phase can be viewed using the Transactions and Others tabsheets of Enterprise View Builder or Enterprise View Tech.

At the end of the analysis the following information can be viewed:

- Transaction – Program relationship
- The type of the relation

This information is available in Enterprise View Tech client solution. By means of the References and Browse Object functions, it is possible to identify in the source code the line that generates the relationship.
18. Enterprise View Analysis Types for IMS Programs

18.1. Introduction

The main elements in IMS application programs are:
- Program entry
- Program communication block (PCB) or application interface block (AIB)
- Input/output (I/O) area definition
- DL/I calls or DB2 calls
- Program termination

The analysis of the programs running in IMS environment can be basic or extended and depends on the programming language in which the programs are written. Therefore different elements of the program are analyzed and different measures and relations are recorded in the Enterprise View Repository and Database. The analysis type of every object in Enterprise View classifications can be viewed in Environment Settings of the Enterprise View Configuration Manager module.

The type of analysis of every programming language that Enterprise View supports for IMS environment will be examined in the next chapters.

18.2. Basic Analysis

Basic analysis is performed for:
- IMS Batch Assembler programs
- IMS Batch C/C++ programs
- IMS/TP Assembler programs
- IMS/TP C/C++ programs

The Enterprise View parser scans the programs during the analysis phase and loads in the Enterprise View Repository the following information:
- Sizing Measures:
  - Lines of Code - LOC
  - Source Lines of Code - SLOC
  - Comment Lines of Code - CLOC
- LOC Change Traffic Measures
- Objects Change Traffic Measures

This information is available in Enterprise View Builder and Enterprise View Tech.

Object references and quality measure information is not recorded. Analysis of the statements in the sources and deep analysis is not performed.
18.3. Extended Analysis

Extended analysis is performed 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

The Enterprise View parser scans the programs during the analysis phase and loads in the Enterprise View Repository the following information:
- Sizing Measures:
  - Lines of Code - LOC
  - Source Lines of Code - SLOC
  - Comment Lines of Code - CLOC
- LOC Change Traffic Measure
- Objects Change Traffic Measures
- Object Reference Analysis for all objects
- Quality Measures
- Deep Analysis results

This information is available in Enterprise View Builder and Enterprise View Tech and can be viewed by using the Programs tabsheet.

At the end of the analysis the following information can be viewed:
- Programs – include books relationships
- Programs – databases relationships
- Programs – called/caller program relationships
- Programs – maps relationships
- The types of the relations

This information is available in Enterprise View Tech. By means of the References and Browse Object functions it is possible to identify in the source code the line that generates the relationship.

For more information see COBOL Process Guide and PL/I Process Guide.

18.4. Analysis Disabled

Analysis is not performed for:
- Map definitions
- MFS map definitions
- IMS transactions