Contents

CHAPTER 1 HP EXSTREAM INTRODUCTION

LEARNING OBJECTIVES: At the end of this chapter, the reader would be able to understand

- *HP Exstream capability and its Architecture.*
- *HP Exstream Components*
- *Operating System supporting HP Exstream*

1.1 What HP Exstream Can Do?

HP Exstream combines the most advanced technology for personalized document creation, campaign management and tracking, and multi-channel delivery to empower businesses to quickly deliver critical documents and relevant information to customers. Output can be prepared for delivery across virtually every print and electronic channel. We can track each communication along with customer responses so companies maintain a meaningful, ongoing correspondences with hundreds or millions of customers.

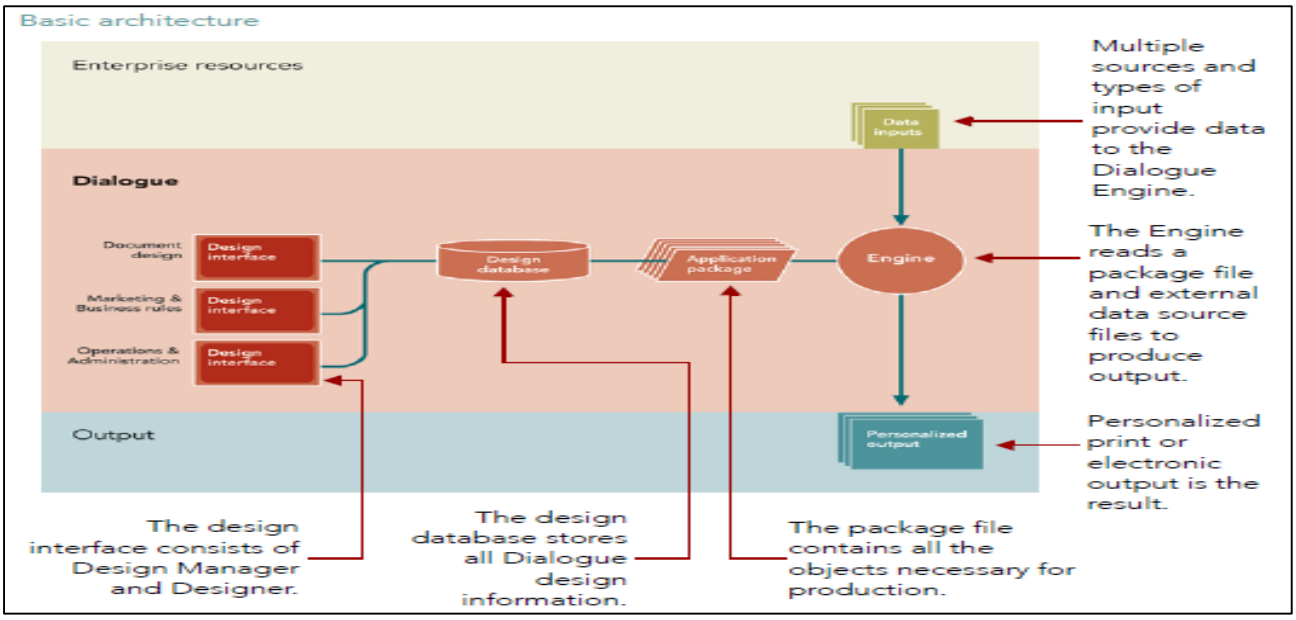
HP Exstream uses customer data, interactions, and preferences to create relevant, personalized communications. HP Exstream's client/server approach enables many people to work on the same application at once. With the software's intuitive graphical interface, designers and business users can develop personalized documents and web applications, create business rules, and define production characteristics.

HP Exstream provides multiple benefits to Customers providing highly modular, scalable and high-volume processing capabilities for an organizations Customer Communication Management needs. In addition to create personalized, value-added documents HP Exstream streamlines the related processes and integrates workflows, eliminating point solutions, pre- and post-processes, redundant activities, and expensive programming and maintenance. Some of the benefits that HP Exstream offers are –

✓ Design and create any type of document—from a simple letter to direct marketing to the most complex statement or portfolio
✓ Support collaboration from users of many skill levels in many different roles—from IT to marketing to operations and other lines of business
✓ Retrieve, process, and update data from multiple systems in many supported file formats providing a great level of architectural compatibility.
✓ Reuse content across the enterprise to streamline processes and ensure consistency
✓ Produce documents on almost all computer platforms—as well as multiple ones—for distributed processing
✓ Output variable documents, whether in batches of hundreds of thousands or one at a time, at very high performance levels, regardless of the delivery channel

1.2 Architecture

HP Exstream's architecture allows the greatest degree of communication flexibility. HP Exstream is designed to be the single connection between enterprise resources and customer communications.



The Design Database

HP Exstream uses ODBC-compliant databases to contain:

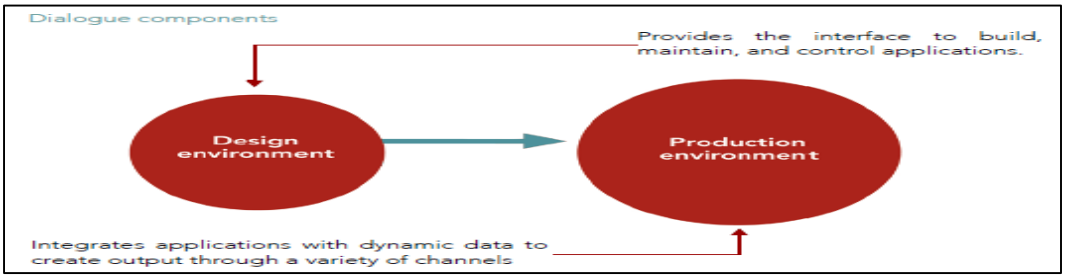
- Information about the objects created, modified, or deleted in Exstream.
- System settings and configuration information.

This design database allows enterprise systems to interact easily with HP Exstream's open architecture.

1.3 HP Exstream Components

HP Exstream operates in two basic environments:

- **Design Environment -** The features of HP Exstream used to design and manage customer communications.
- **Production Environment -** The features of HP Exstream used to produce the final customized customer communications.



HP Exstream's design environment enables us to create page layouts using the intuitive design interface. Graphic design toolbars and menus enables us to compose and create personalized

communications with ease. Once fully tested and packaged, applications are moved to the HP Exstream production environment. The HP Exstream Engine links to systems or databases to retrieve data in real time. HP Exstream can output to multiple devices and in many different formats.

1.3.1 Design Environment

We create communications within a networked design environment. In this environment, there are two programs:

- **Design Manager** - A management software which we use to create and manage the objects in the design database. The design database builds communications, manages Exstream users and security, and defines the delivery environment (for example: fonts, colors, outputs, and mailing systems).
- **Designer** - A graphic design software which we use when designing objects, such as pages, to send to customers.

The design environment must be installed on the Windows workstation where we create the package files. These files are then transferred to a workstation with the production environment installed.

1.3.2 Production Environment

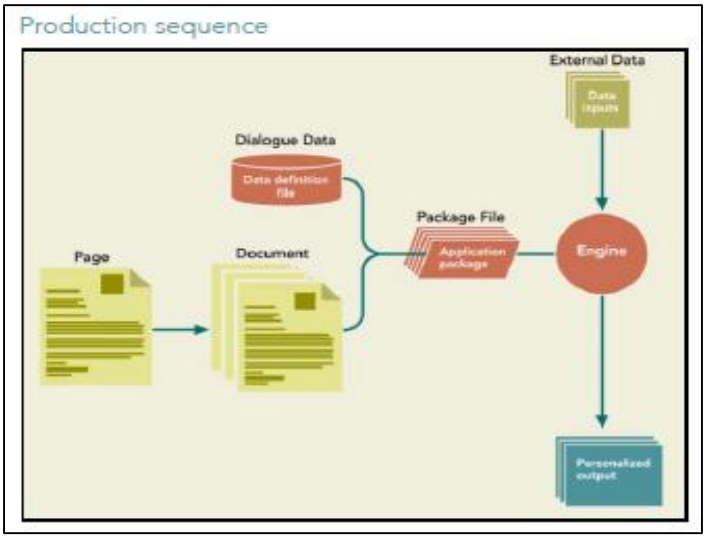
We deliver output via the production environment. Machines with the HP Exstream production environment installed produce output using the Exstream Engine and files created in Exstream.

The HP Exstream Production Engine

The Engine is a stand-alone program that builds personalized communications for customers. It is a batch, transaction-oriented program that can run on various operating systems.

The Design to Production Sequence

The following diagram provides a sample procedure for creating output using the essential objects in Exstream.



To create personalized output using HP Exstream:

1. Create and map a data file.
2. Create and define a page or pages and other application objects. Add design objects to the page.
3. Create and define a document. Add the page(s) and (optionally) message(s) to the document.
4. Create and define an application. Add the document and (optionally) campaign(s) to the application. Add the data file to the application.
5. Build a package file from the application.
   - Packaging an application takes place with Design Manager.
   - The package file contains all the objects needed for an Engine run.
6. Run the Engine using the package file and external data.
   - The package file may be run in Windows for testing and actual production, or transferred to another platform for production.

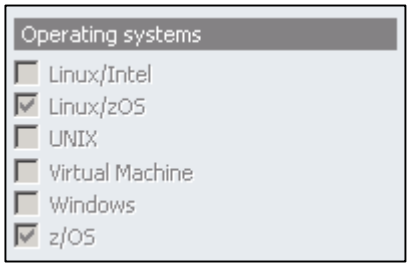
1.4 Operating Systems Options

Once an application is fully designed, tested, and packaged, it is then packaged in the production environment.

To view the Operating Systems on which the Key enables us to run the Engine, go to the Operating Systems area on the Key tab of the System Settings. It is processed with live data by the HP Exstream Engine. Production environments supported by Exstream are:

Windows 98/2000/XP	UNIX	MVS
AS/400	Linux	z/OS

In the example below, this system can utilize Linux and z/OS operating systems.



1.5 Suites and Modules

HP Exstream is packaged in suites and modules that correspond to specific capabilities. The base functionality can be easily expanded to meet our on-demand interactive communication by purchasing additional suites and modules.

We can verify the suites and modules available on the current database and workstation in two locations:

- The ***Key*** tab on System Settings
- The ***Key*** tab on System Configuration

CHAPTER 2 HP EXSTREAM DESIGN MANAGER OVERVIEW

LEARNING OBJECTIVES: At the end of this chapter, the reader would be able to understand

- *Overview on Application, Document and Pages*
- *Basics of Data Files and Data Mapping*
- *Various Output Queues*
- *Variable Properties*
- *Framing Rules*
- *Defining Functions and Subroutines*
- *Sections and Paragraphs*
- *Defining and Using Search Keys*
- *Creation and Definition of Campaigns and Messages*
- *Defining and using Barcodes*
- *Environment objects*

2.1 APPLICATIONS, DOCUMENTS AND PAGES

2.1.1 Introduction to Application, Documents and Pages

Applications, documents and pages are the basic design objects necessary to create personalized output in HP Exstream. Developer needs to create these objects in Design Manager in almost any order before combining them into an application.

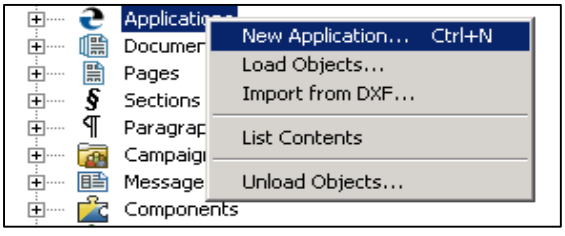
Benefits

Applications and documents are used as containers to which a smaller object can be placed and properties are set to control the selected objects for personalizing a customer.
Pages are used as blank canvases on which different designs can be created.

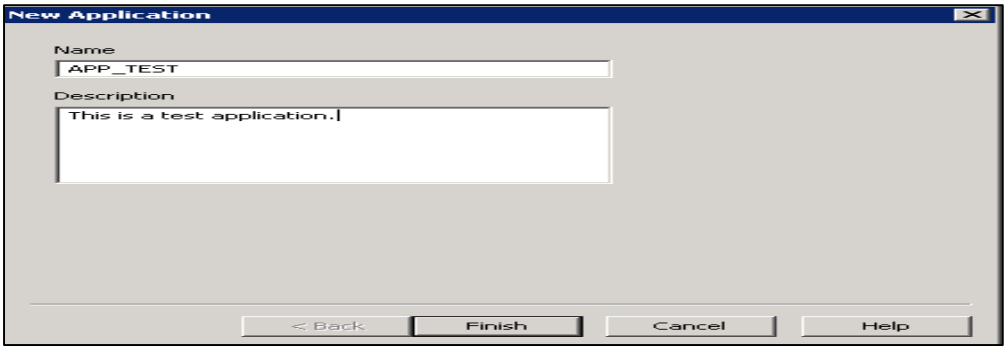
2.1.2 Creating and Defining Applications

An application is a container that holds all the objects needed for packaging to create a personalized communication. Application can be as simple as a single page, a document and a data file, or much more complex by addition of other design objects also.
To create an application in Design Manager:

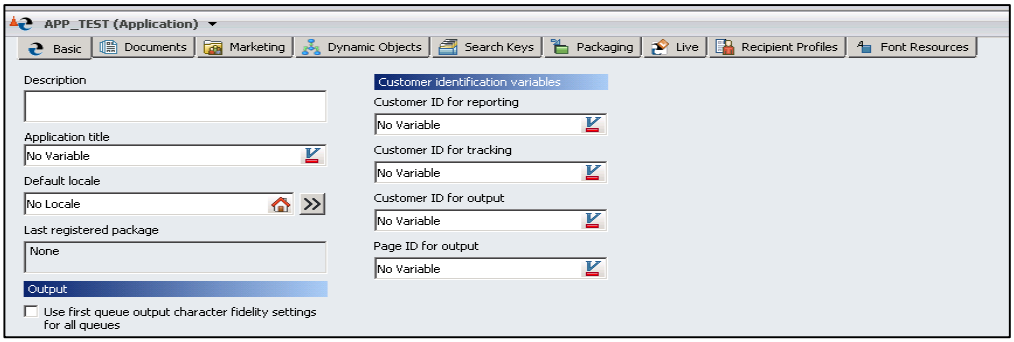
1. In the library, right click the Applications heading and select **New Application**.
   The **New Application** dialog box opens.



2. Enter a name, in the **Name** box. Enter a description (optional), in the **Description** box.
3. Click **Finish.**



The created application is added to the Applications list in the Library and opens in the Property Panel to be defined.



2.1.2.1 Adding Objects to an Application

Minimum requirements of an application to create an output are:

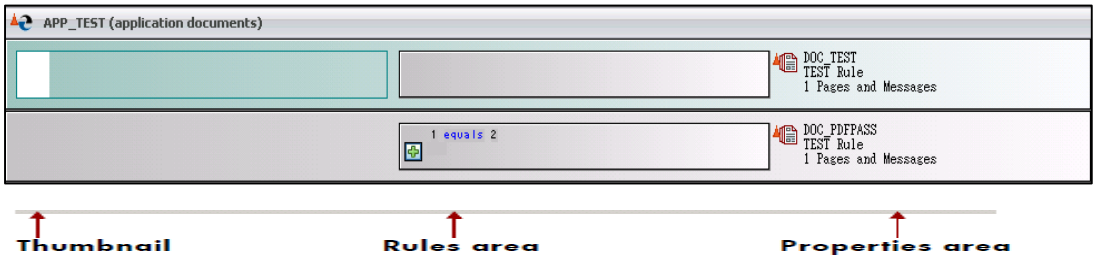
- A document containing at least one page
- A customer driver Data file

NOTE: *Output queue is not required for viewing the designed output in Exstream, it can be viewed using Exstream viewer.*
But in production environment it is mandatory to create an output queue for printing the output or for online viewing or for sending mails.

2.1.2.2 Using Applications

Opening Applications in the Edit Panel

- Applications are container for many objects like documents and campaigns. When an application is dragged to edit panel it prompts to choose the objects which are present inside the application.



- Select the type of object to edit and click **OK.**

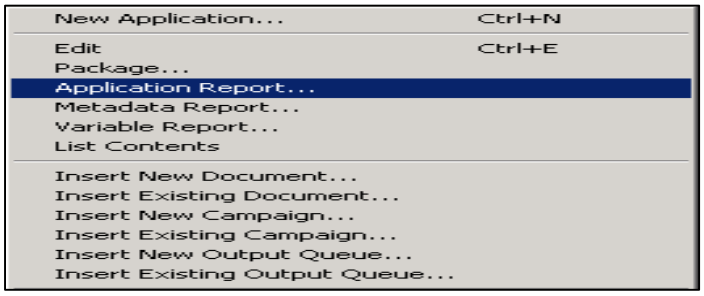
- If single object is present inside the application then on dragging the application to edit panel the object i.e. document or campaign directly opens in edit panel.
- If many objects are present, then double click on the object to open it in edit panel.
- To view the content of an application, right click the application and select **List Contents** from shortcut menu. A list of the contents of application opens in edit panel.

Creating Application Reports

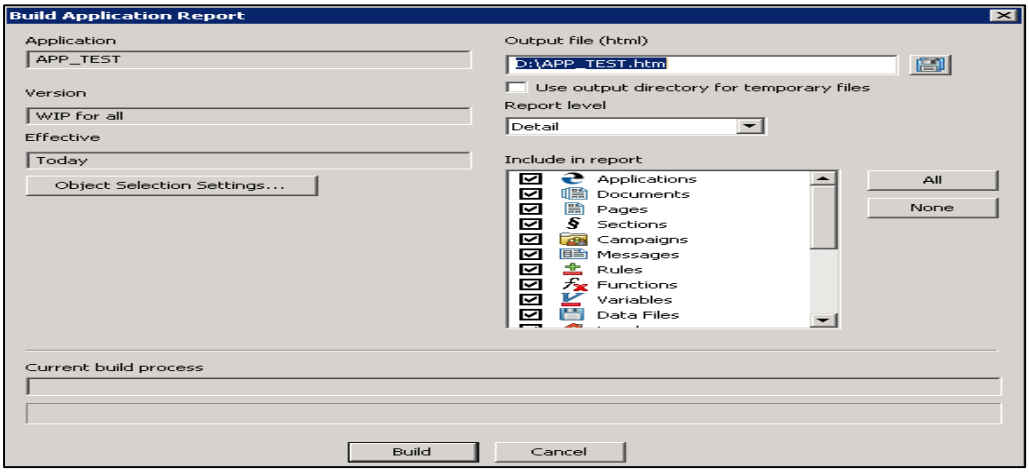
- It is created to see a list of all objects in the application, in HTML format.
- It is useful for troubleshooting, since it provides detailed information about the selected objects.

Building of Application Report

- Right click on Application and select **Application Report**.



- The boxes show the name of the application and the version selection settings.
- In the **Output file (HTML)** box, enter the name of the HTML file that will contain the report. Specify the full path if required to be created in other location else it will be created in a default path inside HP Exstream under the program files in C: drive.
- **Report Level**, from drop down select either **Summary** or **Detail**. **Detail** provides all possible information of each object and generates a large report, but takes longer to run. **Summary** includes less information.
- **Include in Report,** select the object to include in report or else **All** button to include everything and **None** to clear the selected objects.
- The Application report can be viewed using the **Build** button. The report generates in specified location in HTML format.



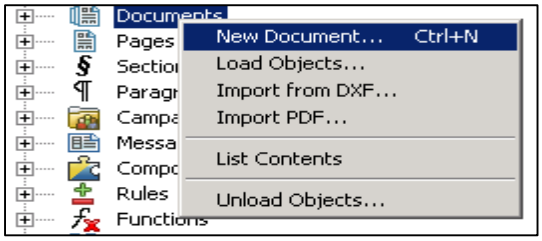
2.1.3 Creating and Defining Documents

Documents are containers for pages, messages, and sections. Document objects are added to the Applications in the Library for reference.

2.1.3.1 Creating a document in Exstream

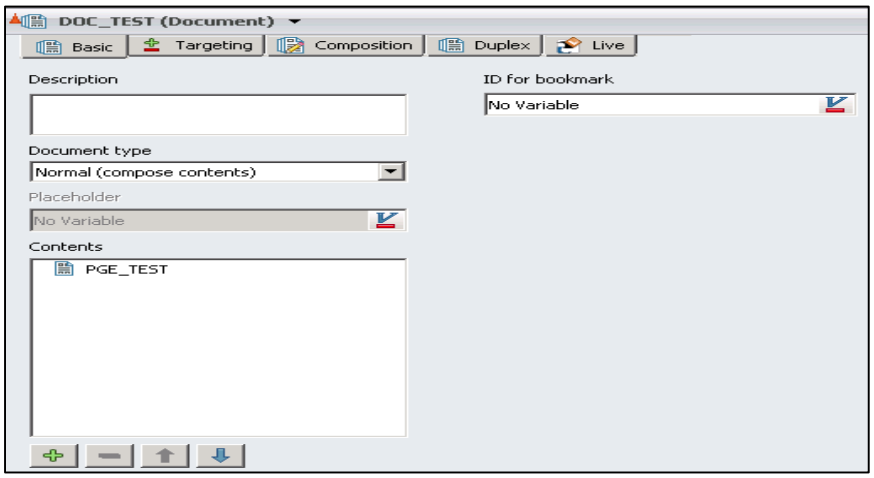
To create a new document:

1. In the Library, right-click the **Documents** heading, and from the shortcut menu, select **New Document**. The **New Document** dialog box opens.



2. In the **Name** box, enter a name and in the **Description** box, enter a description (optional).
3. Click **Finish**.

The new document opens in the Property Panel for us to define.



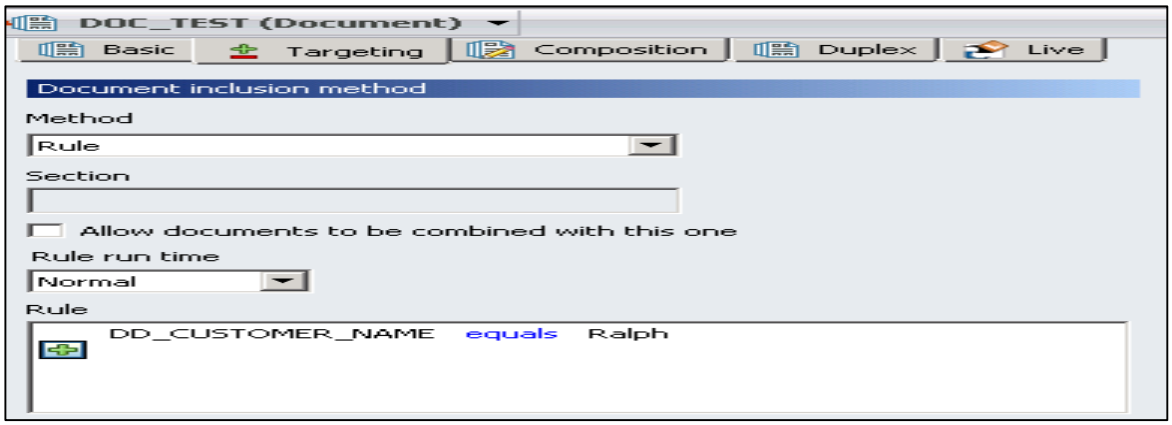
Note: *At least one page to a document must be added to be a valid document. Addition of messages and sections to a document can be done.*

2.1.3.2 Controlling How Documents are Included

Document properties are used:

- To specify how documents are included in an application.
- To control which documents are included for specific customers by creating an application that contains all possible documents and then using section data or rules to control the inclusion of a document.

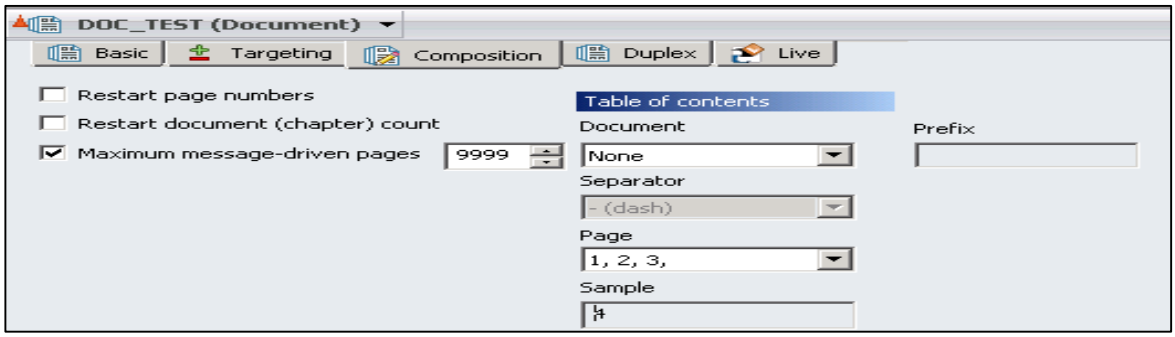
Consider the below snapshot where a document from an application will be triggered if the **Customer Name** in the input file is "**Ralph**".
In this case the triggering rule should be, if the variable representing the customer name is equal to Ralph then the specific document will be triggered and included in the processing.



2.1.3.3 Controlling Page Numbering

To control the page numbers in document the Composition Tab properties are changed.

Force the page numbering to restart for each document in the application (by default, the page numbering continues throughout the application)	Select the **Restart page numbers** check box. Selecting this option affects only the 'SYS_PagePrintedValue,' 'SYS_PagePrint,' and 'SYS_PageTotalPrinted' values.
Display page numbers in tables of contents and indexes from multiple documents as if they were chapters in the same book (for example, if this document were the third document in the application, pages would be numbered 3-1, 3-2, 3-3, and so on)	Select the **Restart document (chapter) Count** check box.



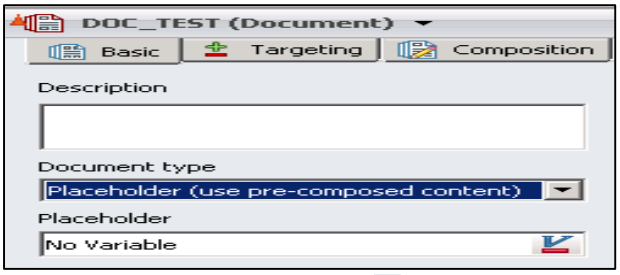
2.1.3.4 Setting Up the Document to be a Placeholder Document

- A document can be used to import the dynamic content into the design at run time.

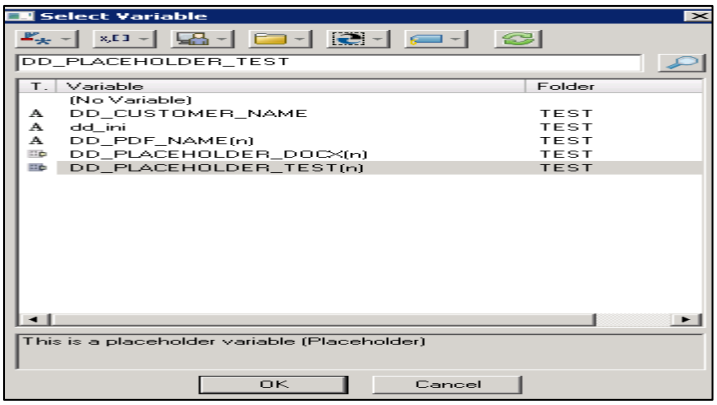
- This needs the license for Dynamic Content Import Module.
- Multiple pages can be inserted into a document without including flow pages in the page object.

To set up the document to be a placeholder document (Pass- through):

1. On the **Basic** tab of the document properties, from the **Document type** drop-down list, select **Placeholder (use pre-composed content)**.
   The **Placeholder** box below becomes active.



2. In the **Placeholder** box, click . The **Select Variable** dialog box opens.
3. Browse to the placeholder variable that controls the placement of the document content and click **OK**.

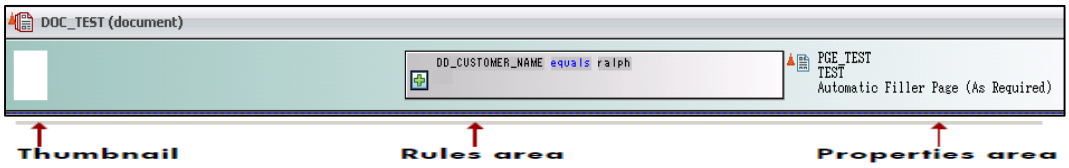


4. The variable appears in the **Placeholder** box.

When the application runs, the content specified by the placeholder variable is imported into the placeholder document.

2.1.3.5 Opening Documents in Edit Panel

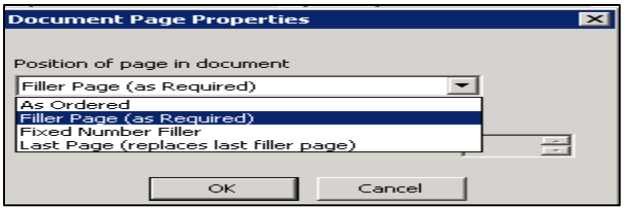
- Drag a document to the Edit Panel, the pages and messages it contains are listed along with thumbnail views and other information such as targeting rules and flow properties.



- Add, delete and changes to the order of pages and messages are possible in edit panel.

- Add, remove, or edit selection rules for each page in Targeting area.
- To add another page or message to the document, drag it to the Edit Panel. A reference to the page or message is added under the document in the Library.
- For setting up a document as a placeholder document, the position of Page in document should be “Filler Page”. If we don’t check this option only first page will be imported.

  To change the Document Page properties, double click the Properties area and select “Filler Page (as required)” option from the drop down.



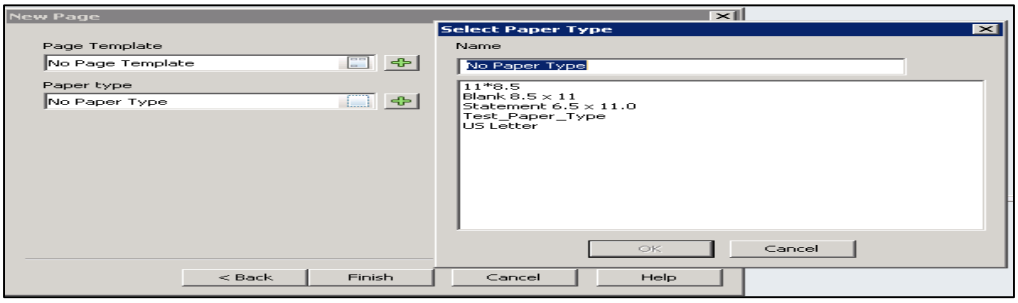
2.1.4 Paper Types and Pages

Necessary objects to define the appearance of a Page are-

- Paper Type
- Pages

Paper Types

- Paper types are necessary for printing a correct page.
- Paper types are also used to determine the size of the page when we open it in Designer.
- Must create a paper types before creating pages or templates.
- Paper type is specified while creating Pages.



Pages

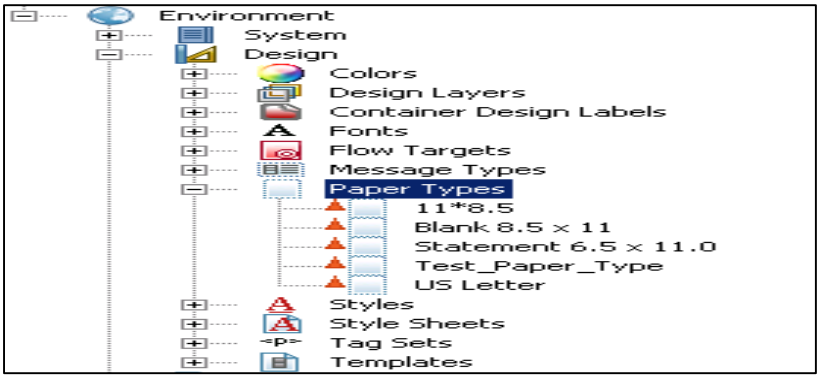
- All Pages are created under the Pages heading in Library.
- Different types of pages are created to serve various purposes. Those are-

DOCUMENT	This type of page is placed directly into the document. Document pages are always available to send if the document is sent. Document pages hold business content.
MARKETING	This type of page is an additional page included in a document if a campaign adds marketing messages that cause Design Manager to include the page when the engine is run.
FLOW	This type of page holds any information that does not fit onto the page in which it was originally placed. A flow page must be designed with a flow frame. Also these pages as flow pages are designed in the Edit Panel.

2.1.6.1 Creating Paper Types

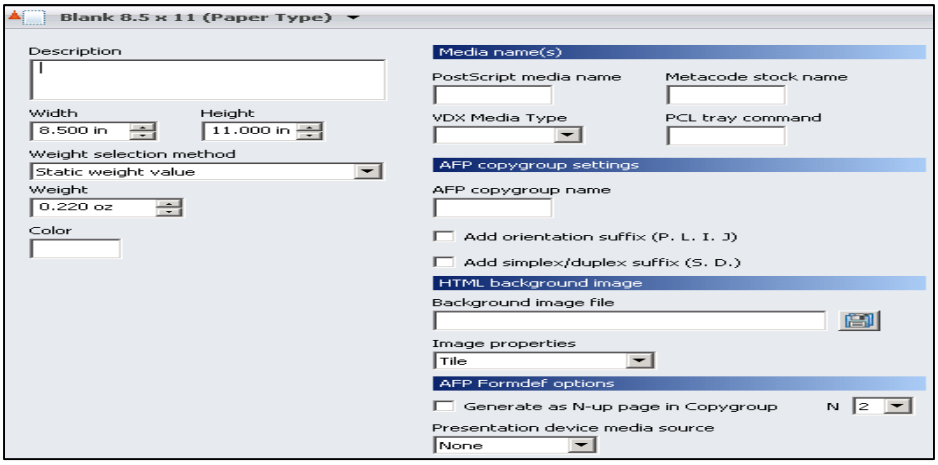
To create a new paper type:

1. In the Library, expand the **Environment > Design** heading.



2. Right-click the **Paper Types** heading, select **New Paper Type**.
3. In the **Name** box, enter a **name** and **Description** (optional) inside **Description** box.
5. Click **Finish**.

The paper type opens in the Property Panel.

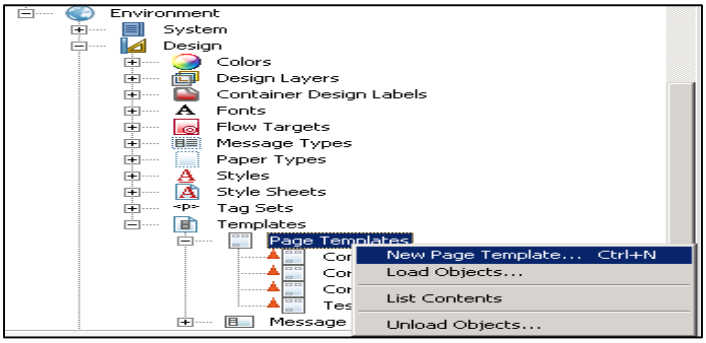


2.1.4.2 Creating Page Template

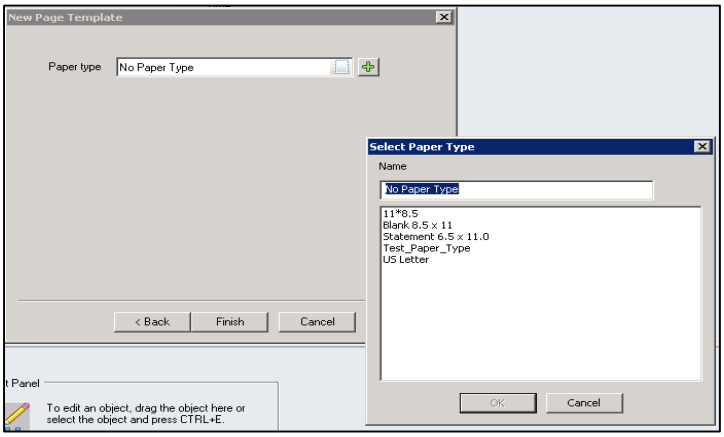
An object that defines a paper type and controls what type of page can be used.

To create a template in Design Manager:

1. Under **Environment→Design→** select the **Templates** heading.
2. Right-click **Page Templates→** New Page Template**→** New Page Template dialog box opens.



3. In the **Name** box, enter a name. In the **Description box**, enter a description (optional).
4. Click the **Next** button.
5. Select a **Paper type** from the drop-down list.



6. Click **Finish**.

The template is created and opens in the Property Panel for us to define.

The properties of template (Basic Tab) are similar to that of Page as mentioned below.

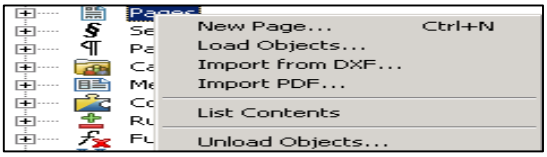
Benefits

Once we design a Page template, we can reuse it in many pages. It also makes the design simple by designing the repetitive parts of a page in a separate template and uses it.

2.1.4.3 Creating Pages

To create a new page:

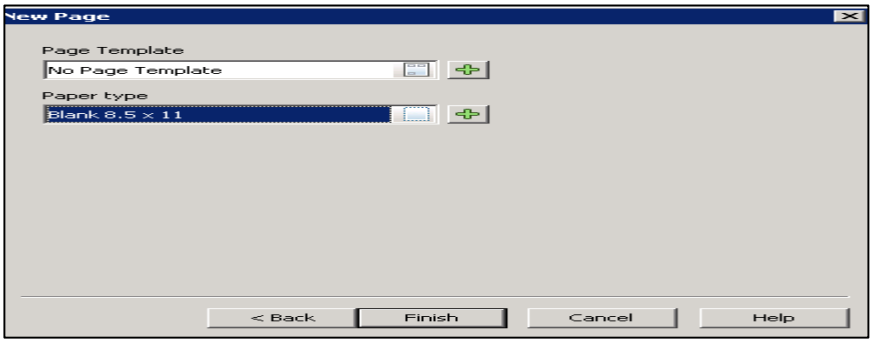
1. In the Library, right-click the **Pages** heading, select **New Page**.



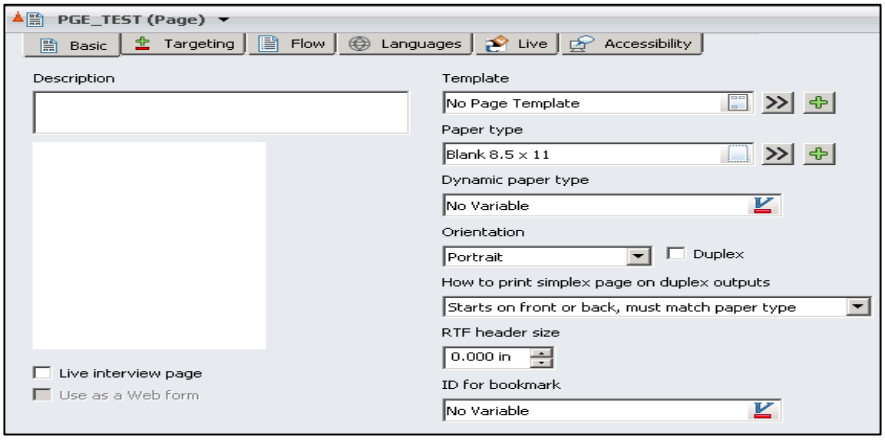
2. In the **Name** box, enter a name. In the **Description** box, enter a description (optional).
3. Click **Next**.
4. From the Page Template box, click and select a page template (optional).

Note: *If we select a page template, the **Paper type** box becomes inactive.*

5. From the **Paper type** box, click and select a paper type. Click **Finish**.



The page opens in the Property Panel.

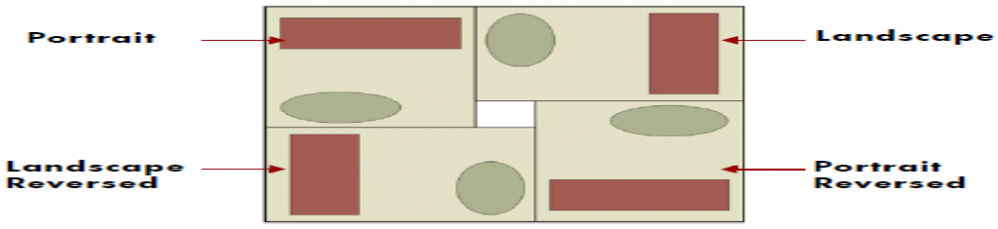


2.1.4.4 Defining Pages

- **Basic Tab:**
  - **Description:** Enter text to differentiate the page from others.
  - **Live interview page*:*** Select the check box to include the page in an interactive document so it can be viewed in the LiveEditor.
  - **Template:** Click to select required template from the list. Templates are created under **Environment** -> **Design** -> **Templates.**
    Click to view or edit the properties of template.

Note: *If we select a template, all other options become inactive.*

  - **Paper Type:** Click *to* select paper type or modify it if already selected.
  - **Orientation:** To change the layout of the page with respect to the printer's normal output page orientation.
    1) **Portrait**: the page is taller than it is wide. The objects on the page print as ordered.
    2) **Landscape**: the page is wider than it is tall. The objects on the page print as ordered.
    3) **Portrait Reversed**: the longer side of the page runs from top to bottom. The objects on the page appear in reverse order.
    4) **Landscape Reversed**: the longer side of the page runs from left to right. The objects on the page appear in reverse order.

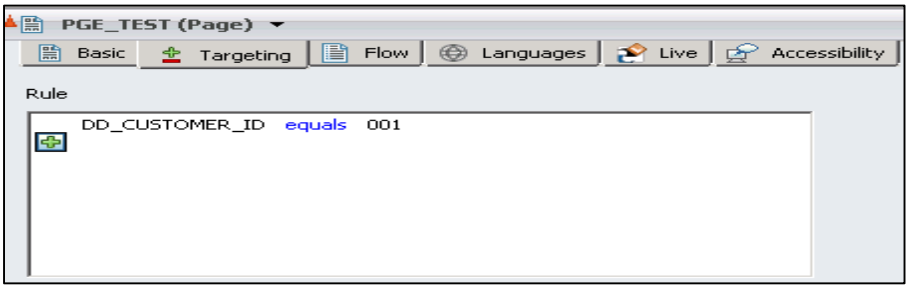


  - **Duplex:** To print the pages on both the sides, select Duplex check box. The content in the page will be printed in both front and back side. If duplex option is selected then **How to print simplex page on duplex outpu**t becomes inactive.

- **RTF Header size:** RTFs can be imported in the page and any objects above the specified location in the **RTF header size** box will be considered as header on the page.

➢ **Targeting Tab:**

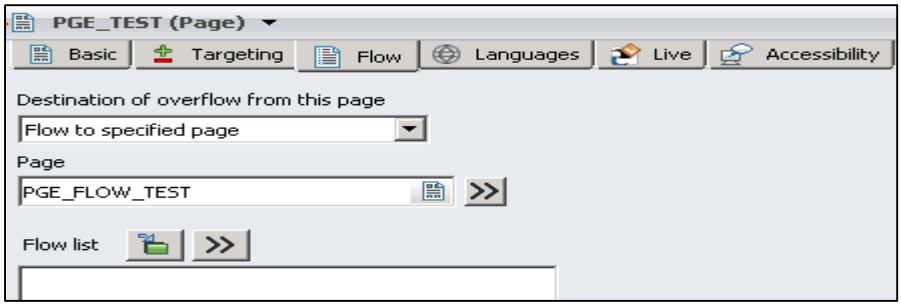
A customer can be targeted or included using rules in targeting tab of page properties. Consider the below snapshot where a page will be triggered if the **Customer ID** in the input file is "**001**". For any other value the page will not be included in the output.



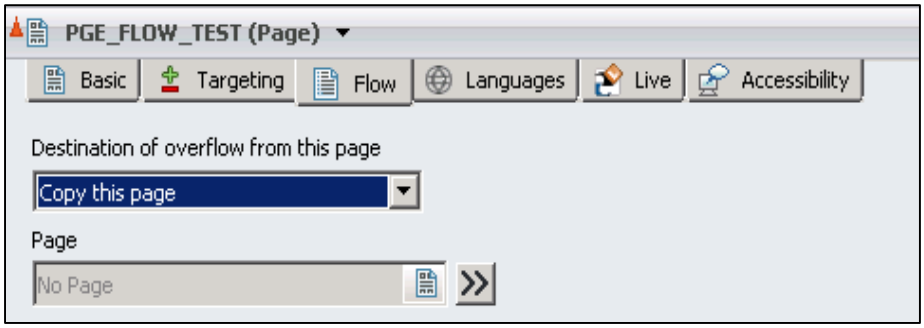
Note: *How to specify a rule in rule dialog box of targeting tab is discussed in **RULE** chapter.*

➢ **Flow Tab:**

- Objects mentioned in the designer like tables and text boxes have the ability to grow based on dynamic information or content. Sometimes the information needs more space than what is allocated in the main page.
- To hold the extra information, **flow frames** are placed in flow pages.
- The flow frame is used as a container which holds the extra content flowing from the main page to the flow page.
- Flow properties needs to be set to enable the flow page.
- Select "**Flow to specified page**" from the drop down of "Destination of overflow from this page".
- In "Page" select the created Flow page name.

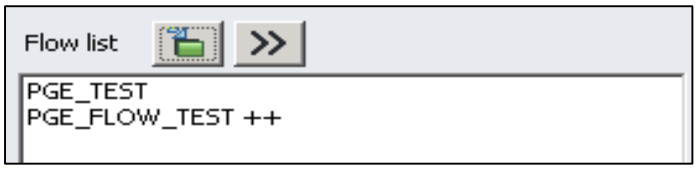


- Save the changes.
- Click  to edit or view the properties of flow page.
- Select "**Copy this page**" option from drop down to keep on copying the same flow page if the content of flow page also exceeds.



- **Flow List:** Lists down all the flow pages linked to the main page. Also the main page is included in the list.

  Click  in the flow tab.



  ++ after a page specifies that the page is set to option "copy this page" and this reproduces multiple times until all flow is placed.

- **Destination of Overflow from this page:**
  This option specifies the position of overflow object from the current page.
  The drop down include options-

  1) **None, ignore overflow**: Overflow is lost. This is the default setting.
  2) **Copy this page**: Overflow is placed on a copy of the current page.
  3) **Flow to specified page**: Overflow is placed on another page. Select the page from the **Page** box.
  4) **Warning, issue message and continue**: Displays a warning when the page is processed. The overflow is lost.
  5) **Error, issue message and stop**: Issues error message and stops processing the application.

NOTE: *A duplex page flows to the back of itself.*

- **Language Tab:**
  A document can not only be delivered to customers in English language but also in native or multiple languages using Language Layer.
- **Live Tab:**
  The **Live** tab is used for interactive document capabilities (Live) of HP Exstream.

2.2 DATA FILE

2.2.1 Introduction to Data Files

Data files objects instruct HP Exstream how to read or write a particular external data source and how that data is to be used by the engine. It defines an overall layout in which a customer data input can be expected along with the corresponding variable mapping and positioning. The engine reads information from a data file through variables which are mapped to the data file object.

Data file objects do not provide data; they always point to existing data sources in the organization (flat file or ODBC databases) that the engine can reach by path and name.

Benefits

Data files form an input framework which will not only define the input format but also can be used as a key to extract other details that are not available directly from customer data. Also with data file objects; we can customize documents for specific customers based on information residing in data sources.
Also it provides dynamic content for customer documents.

2.2.2 Types of Data File

Following are the various types of data files that the engine can use:

Data File Type	Description
Customer Driver File	It contains **customer** data and is required for any application to run.
Initialization File	It is the file that is read first (if present) by the engine to **initialize** values for certain variables.
Reference File	It is basically used by the engine to retrieve that **customer information** which is not included in the customer driver file.
Report File	It is used to create a set of **records** for each customer from customer data as per the requirement.
Sort index File	It is used with output **sorting**. It tells the engine how to **arrange** data fields when it creates the sort index file during pre-processing.
Post-sort Initialization File	It is an **initialization** file that the engine reads during **post-sort** processing.
File viewer	It is used to view **AFP** or **Metacode** output files.
Post sort Report File	This report file is used to create a set of records about each customer from customer data during **Post-sort** processing.
Auxiliary Layout	This file is used to provide **multiple key** variables for use with a reference file or to provide transaction information for use with the MapLayout function.

Some of the important data files have been discussed below.

2.2.3 Data File Sequence

The standard order in which data files are placed as well as processed by the engine in an application is

- **Initialization file**
- **Customer driver file**
- **Reference files**
- **Auxiliary layout files**
- **Report files**

2.2.4 Customer Driver Files

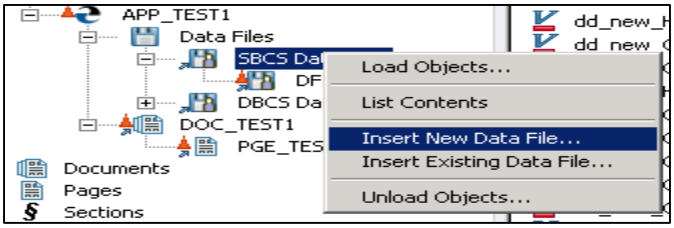
Customer Driver file objects defines the layout of customer-data input. These objects are the only required data file objects in an application.

NOTE: *If an application has multiple customer driver files, the engine processes them in the order in which they appear under the application in the Library.*

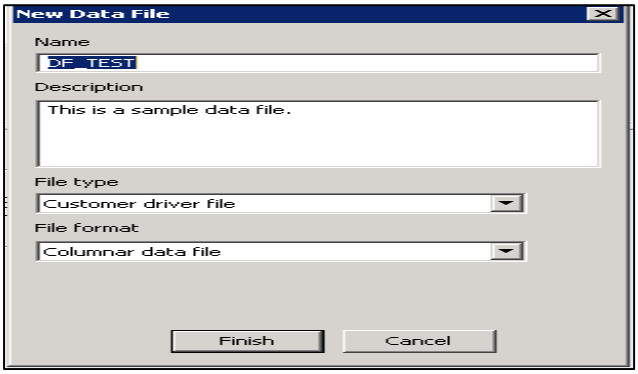
2.2.4.1 Creating Customer Driver Files

To create a customer driver file in Design Manager:

1. In the library, right-click the Data Files heading, and from the shortcut menu, select **New Data File**. The **New Data File** dialog box opens.



2. Enter a name in the **Name** box. Enter a description (optional), in the **Description** box.



3. Select **Customer driver file** from the **File type** drop-down list.
4. In the **File format** drop-down list, we will specify how data is formatted at the external data source. The various options are:
   - **Columnar data file** → The input data will appear in **columns**.
   - **Delimited data file** → A **delimiter character** separates the data.
   - **Print file** →The data appears in a **print** file.

- **ODBC data source** → The data resides in an **ODBC** database.
- **PDF Form** → The data resides in a PDF XFA form.
- **Live** → The data resides in a Live Document (DLF) file.
- **XML data file** → The data appears in an **xml** layout.

NOTE: *If we want to avail certain **File format** options we need to have license to the required modules.*

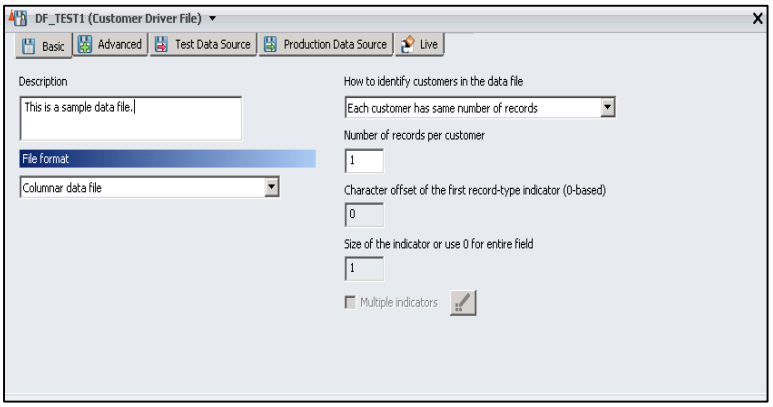
- ***Print File:** Available only if we have licensed the **Print Miner module**.*
- ***ODBC data source:** Available only if we have licensed the **ODBC Access module.***
- ***PDF Form:** Available only if we have licensed the **PDF Form Miner or PDF Form Pre-fill module**.*
- ***Live:** Available only if we have licensed the **interactive document capabilities of HP Exstream (Live).***
- **XML data file:** *Available only if we have licensed the **XML Input module**.*

5. Click **Finish**.

The customer driver file opens in the Property Panel for us to define it.

2.2.4.2 Tabs in Customer Driver Files

- **Basic Tab**



In this tab the basic options for the customer driver file object is set. Following are the various options:

- **Description**: A brief description of the data files usage (optional).
- **File Format:** When the data file is created the file format is already specified from one of the drop-down lists (already mentioned above in **Creating Customer Driver Files**).

NOTE: *For delimited data file, additional options appear in the File format area.*

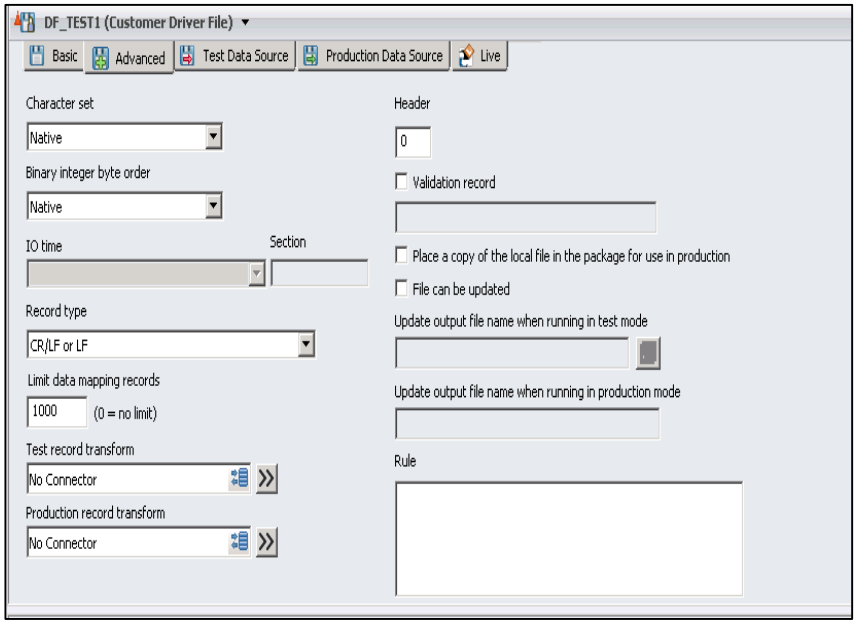
- ***Delimiter** box: The character used to separate data is mentioned.*
- ***Show hex** check box. Selected if the delimiter needs to appear in hexadecimal code.*
- *The **Fields may be quoted with** check box is selected if the delimiter data contains quoted fields. In the adjacent box the character that will surround the data information is specified. The engine ignores this character when it processes the data.*

- **How to identify customers in the data file:** This drop-down list contains various methods to identify the beginning of a new customer. The various methods are:

1. **Each customer has the same number of records-** This is the default option. On selecting this option another box **Number of records per customer** becomes active which also needs to be filled up.
2. **A new customer occurs on specified record type(s)-** This option allows to specify a record type indicator in the data file. On selecting this option the following settings becomes active and needs to be filled up as well.
   - **Character offset of the first record-type indicator (0-based)-** Here we will specify the beginning location of first record-type indicator (columnar and print files only).
   - **Size of the indicator box-** Here we specify the length of the record-type indicator. For delimiter files, the length is specified as 0(as it is automatically determined based on location of next delimiter).
   - **Delimited field index of the first record-type indicator (1-based) box -**It specifies the beginning location of the first record-type indicator for delimited file formats.
3. **A customer ID is on every record**- This option is selected if each file has a unique customer record identifier. When this changes a new customer starts. On selecting this option each of the settings specified in the above option also needs to be filled up here.

❖ **Multiple Indicators**: To specify more than one record-type indicator we use this property. The **Edit Record Indicators** dialog box opens on selecting this option. It is used to specify additional indicators (also referred to as sub-indicators).

➢ **Advanced Tab**



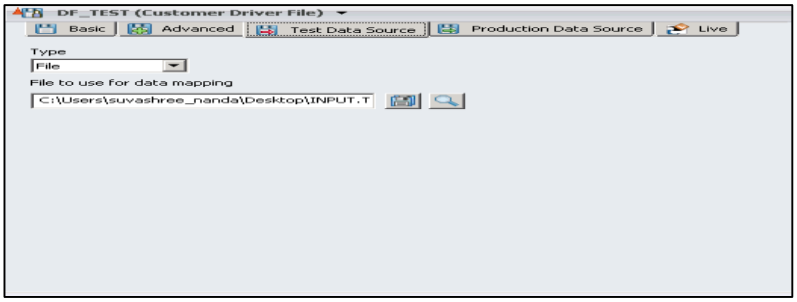
This tab will determine the manner in which data will be read by the engine and processed to generate output. Following are some of the frequently-used fields under the Advanced Tab:

- **Character set**: This is used to identify type of data. Under it we can select one of the following options:
  1. **Native**
  2. **ASCII**
  3. **EBCDIC**
  4. **UTF-8**
- **Binary integer byte order:** This is used to get the required byte order. Under this field we have following options:
  1. **Native**
  2. **Big-endian**
  3. **Little-endian**
- **IO Time and Section:** This field is inactive when a customer driver files is being created. It is used to specify when the engine can read data records from the file.
- **Record Type:** It specifies how records begin and end in the data source. The various options are

1. CR/LF or LF	2. Fixed Length
3. Blocked (prefixed w/ 2 byte big-endian, exclusive)	4. CR/LF only
5. MVS FTP block mode (prefixed w/ x80 + 2 bytes)	6. Blocked (prefixed w/ 4 byte big-endian, inclusive)

- **Limit Data Mapping Records:** This field lets us limit the number of records we view in the Edit Panel.
- **Test Record Transform and Production Record Transform:** We can assign a connector to a data file that points to a routine that transforms incoming data before it reaches the engine. The routine can manipulate the data on a record-for-record basis or in a buffer.
- **Header:** Here we specify the number of records the headers occupy in the file, so that it will not be used by engine to produce output.
- **Rule:** It allows adding a rule to customer driver file object.

- **Test Data Source Tab**



This tab pertains to test runs of the application. Following are the properties set-up in this tab:

- **Type**: The type of data source is selected here. The options are:
  1. **File**: The engine receives data from an external file.
  2. **Connector**: The engine receives data from an external customer-defined routine that is identified by a connector object in the Library.
- **File to use for data mapping**: Here we specify the path to the external file to be used for testing purpose.
- **Connector**: This box is active only when connector **Type** is selected. Here we select an existing connector object in the Library.

- **Production Data Source Tab**

This tab pertains to production runs. Following are the properties set-up in this tab:

- **Type**: The type of data source is selected here. The options are:
  1. **File**: The engine receives data from an external file.
  2. **Connector**: The engine receives data from an external user-defined routine that is identified by a connector object in the Library.
- **File to use for data mapping**: Here we specify the external data source for actual production runs.

2.2.5 Initialization Files

An Initialization file is an optional data file that is used to initialize values to specific variables when the engine runs. It is read once when the engine starts. HP Exstream holds this information in memory until the engine completes its processing.

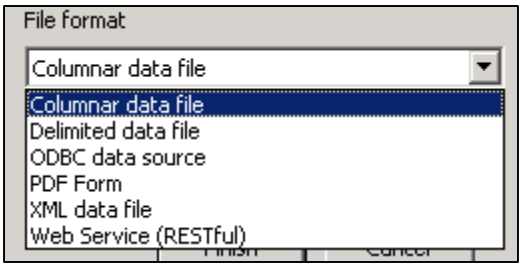
Benefits

This data file object is useful when we require setting values to certain variables, which are customer-independent, just once before the customer-processing.

Following are the steps followed to create an initialization file:

1. Under the **Data Files** heading right-click and from the shortcut menu, we select **New Data File**. The **New Data File** dialog box opens.
2. Enter a name in the **Name** box and a description (optional) in the **Description** box.
3. From the **File type** drop-down list **Initialization file** is selected.
4. From the **File format** drop-down list, one of the following options is selected as per data format.

1. Columnar data file	2. Delimited data file
3. XML data file	4. ODBC data source
5. PDF Form	6. Web Service



5. Click **Finish**.

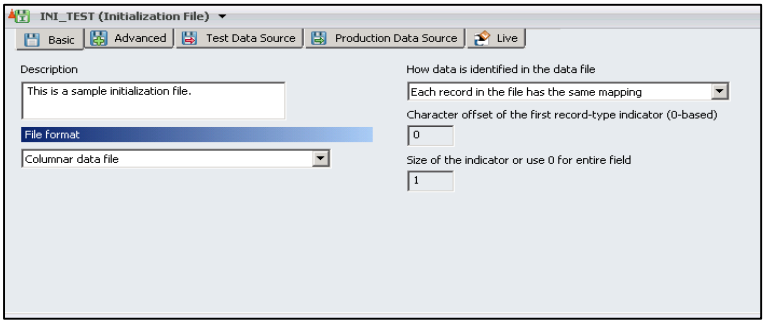
The initialization file opens in the Property Panel so that it can be defined.

NOTE: *Initialization files shouldn't be used when large amount of information is needed to be loaded to memory because it reduces the processing efficiency.*

2.2.5.2 Tabs in Initialization File

The details of each tab have already been discussed in the Customer data file section.

- **Basic Tab**



This tab deals with how the engine interprets and handles data.

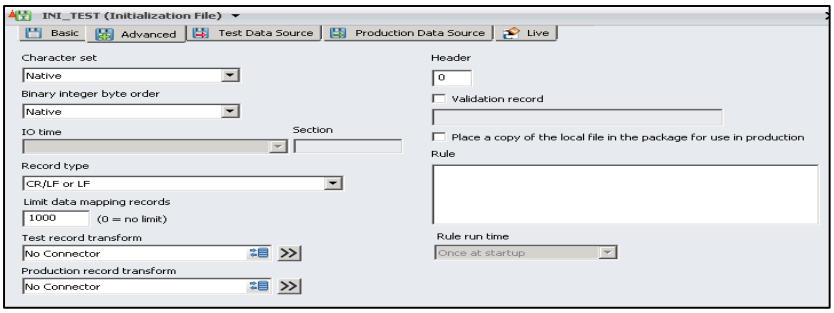
- **How Data is identified in the Data File:** In this drop-down list we select an approach to identify indicators. The various approaches are:
  1. **Each record in the file has separate data mappings:** Each line in the data file has a different mapping.
  2. **File has specified record type(s):** This option allows us to specify the record-type indicator in the data file.

The Start point of the indicator is specified in the **Character offset of the first record-type indicator** field. The length of the indicator is specified in the **Size of the indicator** box.

3. **Each record in the file has the same mapping:** Each line in the file is mapped in the same manner.

NOTE: *We cannot create sub-indicators in initialization files. It can only have one indicator.*

- **Advanced Tab:**



  - **Rule Run Time:** The rule specified in the initialization file under **Rule** can only be read **once at startup** and executed once before file is accessed. If the rule returns exclude then the file is not opened or added to internal file lists.

- **Test Data Source**
- **Production Data Source**

2.2.6 Reference Files

A Reference file is useful when we want to access customer information that is not available in customer driver file. Based on one or more variables (which serve as a key) of customer driver file we can extract other relevant customer data from reference file objects. These are also optional data files placed after customer data files.

2.2.6.1 Creating Reference Files

A Reference file can contain multiple reference files and a reference file can call another reference file. The sequence of reference files must be maintained so that the reference file that refers to another is placed first in the application.

To create reference file object we follow the below steps:

1. Under the **Data Files** heading right-click and from the shortcut menu, we select **New Data File**. The **New Data File** dialog box opens.
2. Enter a name in the **Name** box and a description (optional) in the **Description** box.
3. From the **File type** drop-down list **Reference file** is selected.
4. From the **File format** drop-down list, one of the following options is selected as per data format.

1. Columnar data file	2. Delimited data file
3. Print File	4. Live

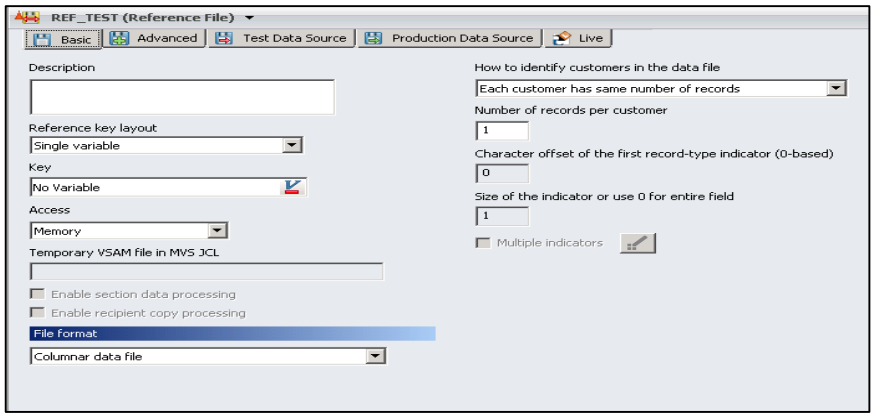
5. XML data file	6. ODBC data source
7. PDF Form	8. Web Service

5. Click **Finish**.

The Reference file opens in the Property Panel so that it can be defined.

2.2.6.2 Tabs in Reference File

➢ **Basic Tab:**



- **Reference key layout:** Using this drop-down list, we specify if a **single variable** or multiple variables in an already created **auxiliary layout file** is used a key to fetch customer-data that is not available directly from customer data file.
- **Key:** This box becomes inactive once an auxiliary layout file is selected. In case of single variable, this box is used to specify the single variable that will be used as a key in the reference file.
- **Access:** From the **Access** drop-down list, one of the following methods is selected for storing and re-accessing information from a reference file.

1. Memory	5. Disk seek
2. Create VSAM	6. Keyed VSAM
3. User routine	7. Driver-ordered, required
4. Driver-ordered, optional	

- **Temporary VSAM File in MVS JCL :** In the Temporary VSAM file in MVS JCL box, we specify the VSAM file name to be used with the **Create VSAM** option in the Access drop-down list.
- **Enable Section Data Processing:** To recognize and read section data in the reference file this check box needs to be checked. This option is available only when the following options are selected in the **Access** drop-down list: **Disk seek**; **Keyed VSAM; Driver-ordered**, **required; Driver-ordered, optional.**

- **Advanced Tab:**
  - **IO Time:** On a reference file object, the IO Time drop-down list lets us specify when the engine will read data records from the file. Following options are available:

1. After initialization files read	7. After initial customer data
2. After each data section	8. After customer data and each section
3. After named data section	9. At the end of each customer
4. Start of queue and queue break	10. Completion of all customers
5. Queue break and end of queue	11. When TRIGGER called from rule/ formula
6. When each page is selected	

  - **Rule Run:** The Rule run drop-down list lets us control when the rule will be processed. Following are its options:
    1. **Once at startup**—Processed only once, before the data file is opened
    2. **Before each operation**—Processed before each lookup
- **Test Data Source Tab**
- **Production Data Source Tab**

For further details on the Advanced, Test Data Source, and Production Data Source tabs please refer to the Customer Driver Files section 2.2.4.

2.2.7 Auxiliary Files

Auxiliary Files are used when we require more than one variable from the customer driver file to be used as a **key** to extract other relevant details that are not available from customer data through reference files. This is also an optional data file placed after reference file.

NOTE: *We cannot add rules to auxiliary layout files.*

2.2.7.1 Creating Auxiliary Layout File

To create auxiliary layout file object we follow the below steps:

1. Under the **Data Files** heading right-click and from the shortcut menu, we select **New Data File**. The **New Data File** dialog box opens.
2. Enter a name in the **Name** box and a description (optional) in the **Description** box.
3. From the **File type** drop-down list **Auxiliary layout file** is selected.
4. From the **File format** drop-down list, one of the following options is selected as per data format.

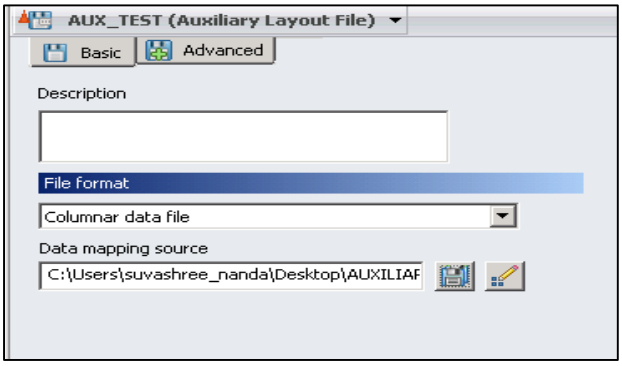
1. Columnar data file	2. Delimited data file
3. XML data file	4. ODBC data source
5. PDF Form	6. Web Service

5. Click **Finish**.

The Auxiliary file opens in the Property Panel so that it can be defined.

2.2.7.2 Tabs in Auxiliary Layout File

- **Basic Tab:**



Here we can change the **File format**. We can specify the external layout file in the **Data mapping source** field to define the key in case of columnar, delimited and XML file formats.

- **Advanced Tab**

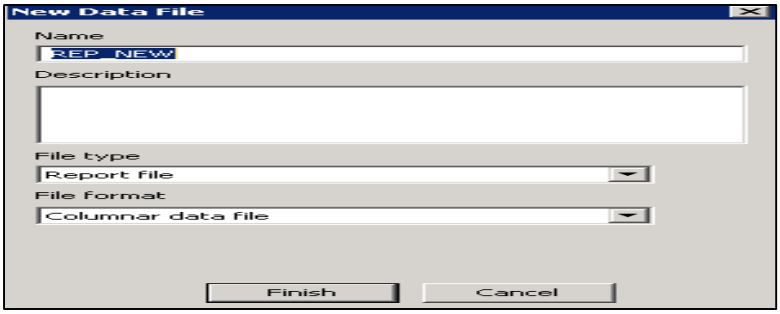
2.2.8 Report Files

This data file is used to create a customized report of relevant information from the customer data as per the requirement. It is also an optional data file that is placed below all data files in the application.

2.2.8.1 Creating Report Files:

To create report file object we follow the below steps:

1. Under the **Data Files** heading right-click and from the shortcut menu, we select **New Data File**. The **New Data File** dialog box opens.
2. Enter a name in the **Name** box and a description (optional) in the **Description** box.
3. From the **File type** drop-down list **Report file** is selected.

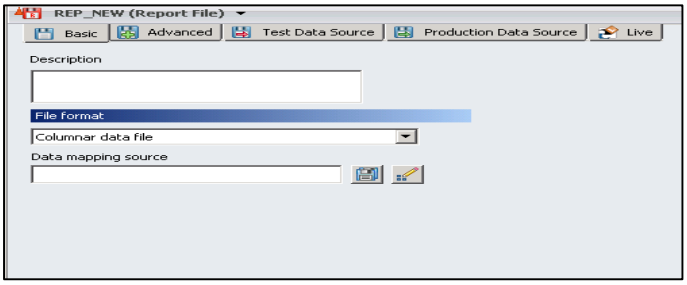


4. From the **File format** drop-down list, we can select one of the types as mentioned under auxiliary file
5. Click **Finish**.

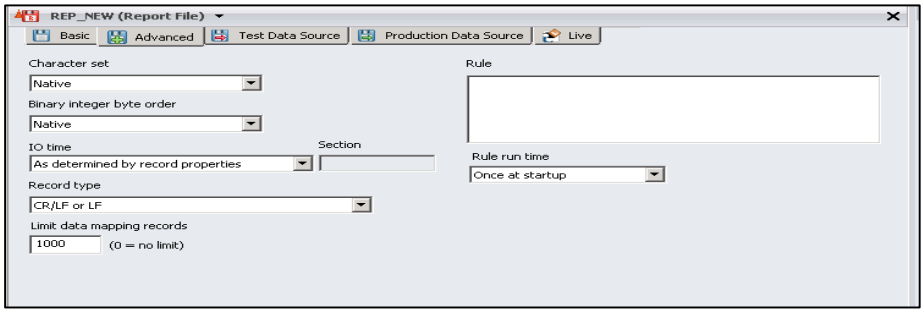
The Report File opens in the Property Panel so that it can be defined.

2.2.8.2. Tabs in Report File

- **Basic Tab**



To use a report file object, we should have a sample layout file for Design Manager to use as a template. We specify a sample in the Data Mapping source field.



- **Advanced Tab**
- **Test Data Source Tab**
- **Production Data Source Tab**

For further details on the Advanced, Test Data Source, and Production Data Source tabs please refer to the Customer Driver Files section 2.2.4.

NOTE: *We cannot use the same file name in the Basic and the Test Data Source or Production Data Source tabs. Else, the output will write over the contents of the layout file.*

2.2.9 XML Data Files

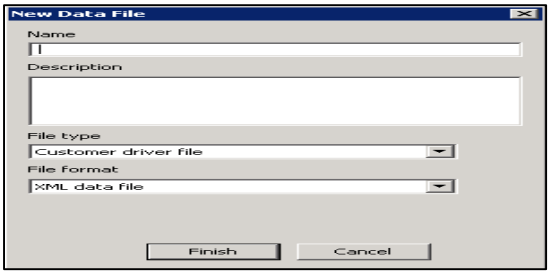
2.2.9.1 Creation of XML Data Files

To create XML data file object we follow the below steps:

1. Under the **Data Files** heading right-click and from the shortcut menu, we select **New Data File**. The **New Data File** dialog box opens.
2. Enter a name in the **Name** box and a description (optional) in the **Description** box.
3. From the **File type** drop-down list one of the following types (which supports xml file format) is selected.

XML Input	XML Output
❖ Auxiliary layout ❖ Customer driver ❖ Initialization ❖ Post-sort initialization ❖ Reference	❖ Report ❖ Post-sort report

4. From the **File format** drop-down list, **XML data file** is selected.

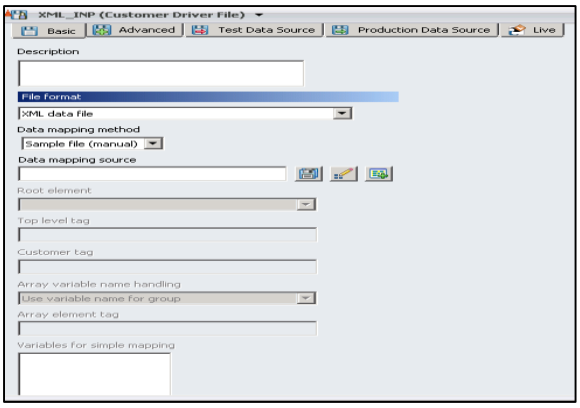


5. Click **Finish**.

The Data File opens in the Property Panel so that it can be defined.

2.2.9.2. Tabs in XML data file

- **Basic Tab:**



  - **Data mapping method:** This option allows us to specify a method to map variables. The methods are

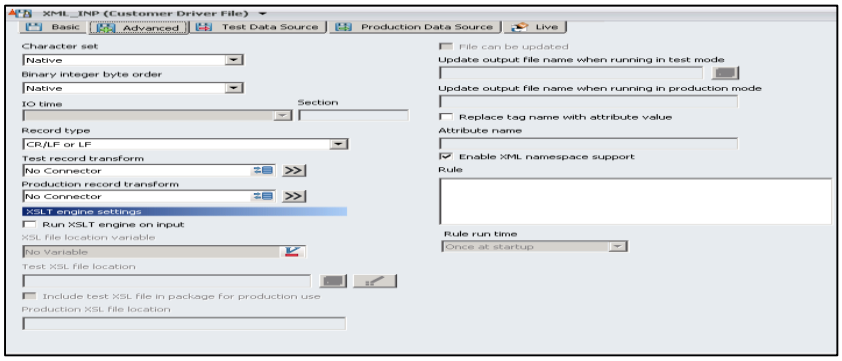
    1. **Manual –** Data file mapped in edit panel.
    2. **Simple-** Design Manager provides a list of variable names.
    3. **Dynamic**—Design Manager maps the data file at run time.

    If we select **Simple** or **Dynamic**, we can use the following options:

    1. **Top level tag**—We enter the top level tag, which is used with most reports.
    2. **Customer tag**—We enter the tag used to indicate a new customer.
    3. **Array variable name handling**— We select the method used to map array variables to tags.
    4. **Array element tag**—We enter the tag to use for the array group.
    5. **Variables for simple mapping**—We add and remove the variables used for simple mapping.

- **Advanced Tab:**
  - **Replace tag name with attribute value:** It lets us determine how Design Manager automaps the data file source. We use this option if we want Design Manager to auto-map a layout file and the existing tag names are generic or are not

descriptive. If we select the **Replace tag name with attribute** value check box, we will also enter an attribute value to replace the tag name in the **Attribute name** box.



- **Test Data Source**
- **Production Data Source**

For further details on the Advanced, Test Data Source, and Production Data Source tabs please refer to the Customer Driver Files section 2.2.4.

2.2.10 Data Mapping

When the engine runs, it obtains data about each customer from at least one data file object. Data mapping lets HP Exstream extract data from a data file. Data mapping lets HP Exstream identify the location of the data, the type of data in each area, and which variable(s) to use to process them.

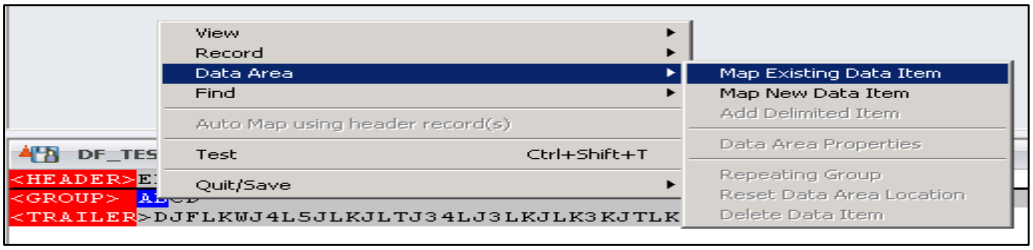
2.2.10.1 Mapping Data Files

To create output, we must map variables to data areas in the data file object. In a data file object that contains one set of data for each customer, the engine reads all the data for the customer and then builds the customer document. This data file does not need to identify the type of records the engine reads (using record type indicators), since all records have the same record format. For example, if we set the file properties on the **Basic** tab to **Each customer has the same number of records**, each record occupies one line.

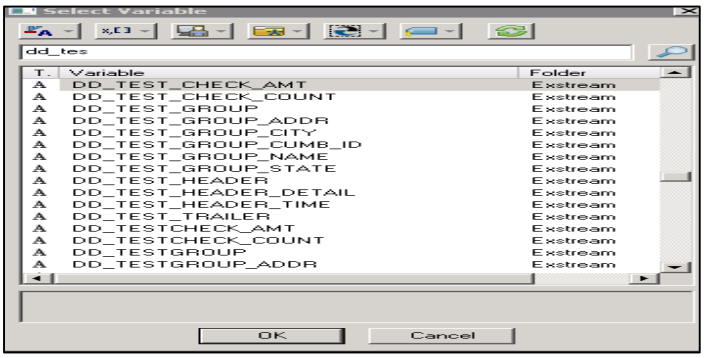
2.2.10.2 Variable Exists

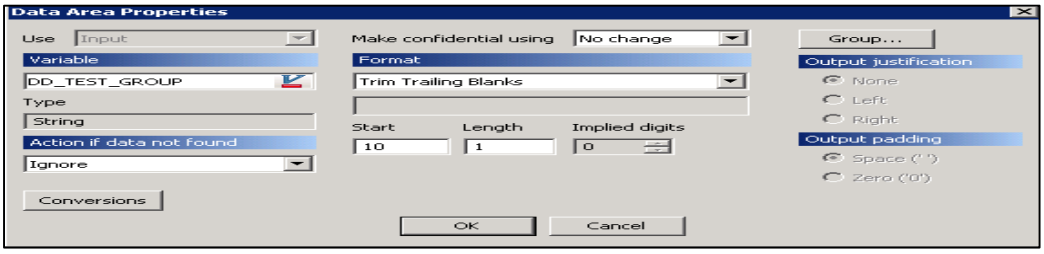
If a data area corresponds to an existing variable, we can map it. To map a data file with an existing variable:

1. Drag the file to the Edit Panel.
2. Go to an unmapped data area we want the engine to access. Unmapped areas are colored gray.
3. Click and drag or double-click to highlight the data area we want to map.
4. Right-click the area and select **Data Area > Map Existing Data Item**. The **Variable panel** opens.



5. Double-click the desired variable. The **Data Area Properties** dialog box opens.





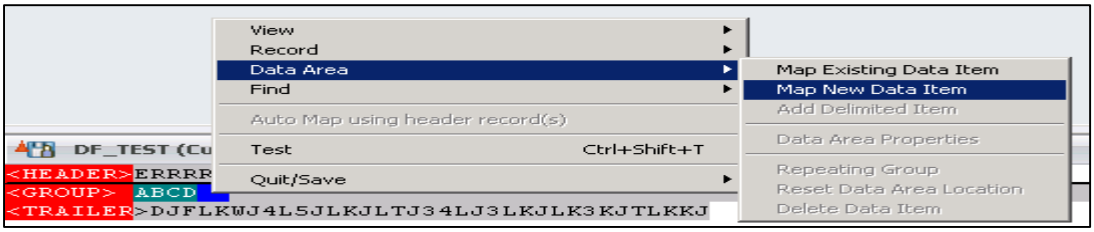
6. Make any necessary changes to the properties and click OK. The **Data Area Properties** dialog box closes. The data area is highlighted in green to indicate that a variable is now mapped in the data file.

2.2.10.3 Variable Does Not Exist

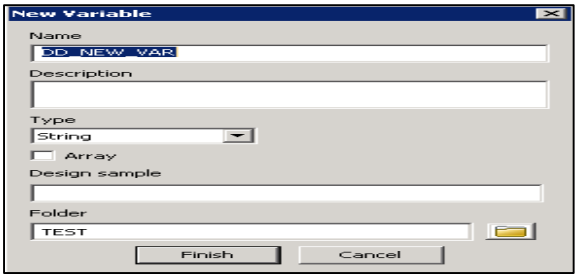
If we are mapping a data area that does not correspond to an existing variable, we must create a variable.

To create a variable to map to a data area:

1. Drag the data file to the Edit Panel.
2. Go to an unmapped data area we want the engine to access. Unmapped data areas are colored gray.
3. Click and drag or double-click to highlight the data area we want to map.
4. Right-click the area and select **Data Area > Map New Data Item**. The New Variable dialog box opens.



5. In the **Name** box, enter a name. In the Description box, enter a description (optional).



6. In the **Type** drop-down list, select a data type.
7. Select the **Array** check box if the variable has multiple values.
8. In the **Design sample** box, enter a sample of the variable text. The highlighted data area is the default.
9. In the Library, specify the folder where we want the new variable to reside.
10. Click **Finish**.

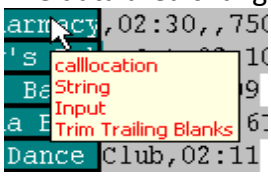
2.2.10.4 Double Mapping

Sometimes we might use the same data to populate two different variables. Rather than having the same information appear in the data file twice, map multiple variables to the same data area.

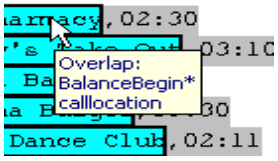
To map a data area that is mapped to another variable:

1. Highlight the data area.
2. Specify the variable to map.

The data area changes from green to cyan to show the area is double-mapped.



Single-mapped pop-up box



Double-mapped pop-up box

2.2.10.5 Delete a Variable Mapping

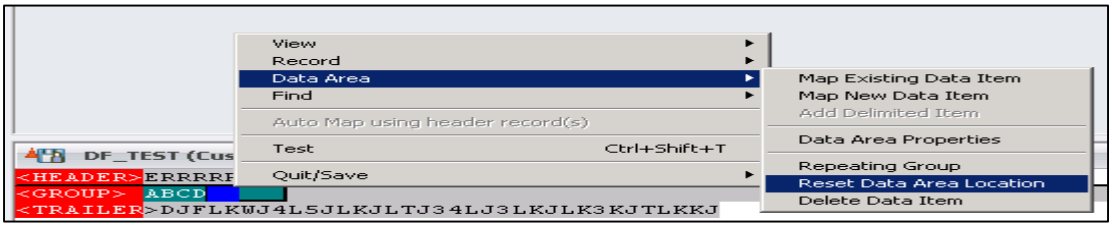
We can unmap a data area without changing the data area itself. Click the mapped area to make it active. A solid line appears around the data area.
There are three ways to unmap the data area:

1. Click the variable name in the layout view, right-click, and then select **Delete**.
2. Right-click and select **Data Area > Delete Data Item**.
3. Click  on the Data Mapping toolbar.

2.2.10.6 Resize a Data Area

We can resize a mapped data area to accommodate changes to the data file object.
To resize a mapped data area:

1. Click the mapped area to make it active. A solid line appears around the data area.
2. Highlight the new length of the data area.
3. Right-click and select **Data Area > Reset Data Area Location**.

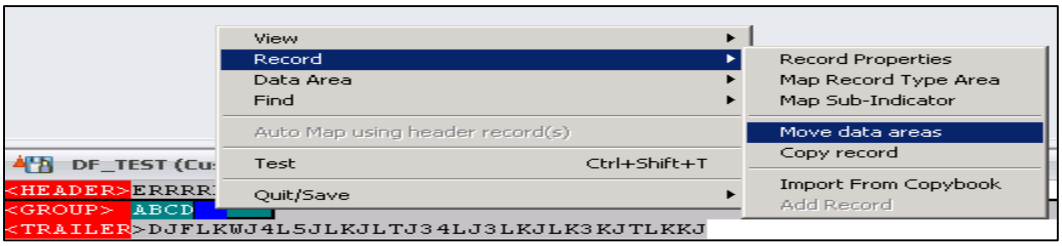


The data area resizes to the highlighted length.

2.2.10.7 Move a Data Area

We can move data areas to insert or delete data items in a record.
To move a data area:

1. Highlight the area we want to move.
2. Right-click and select **Record > Move data areas**. The **Move Data Areas** dialog box opens.



3. In the **Move** box, enter the number of columns we want to move.

4. Select the **Move left** check box to move the data items to the left. If we clear this box, the data items move to the right.
5. Click **OK**.

The data items move the number of spaces we specify.

2.2.10.8 Navigating a Data File

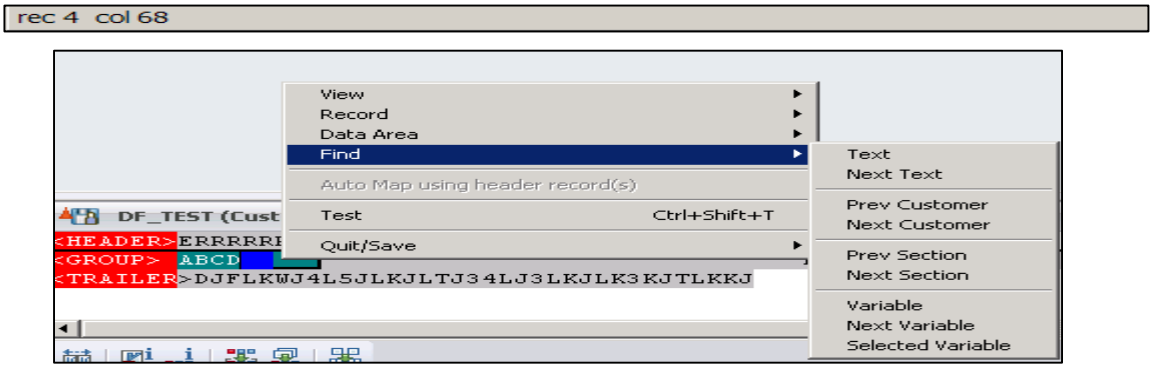
There are several features in Design Manager that lets us quickly navigate within a data file object. These include:

1. The Status Bar
2. Searching for text

3. Searching for a variable
4. Navigating by customer/section

1. Status Bar

As we move the pointer through a data file in the Edit Panel, Record and Column numbers appear in the bottom left corner of the Design Manager window.



2. Searching for Text

We can move quickly through a data file object by searching for specific text.
To search text, right-click in the Edit Panel and select **Find > Text** to open the **Find** dialog box. In the Find box, enter the text we want to find in the data file object.

3. Searching for a Variable

We can move quickly through a data file by searching for a specific variable.
To search for a variable:

1. Right-click in the Edit Panel and select **Find > Variable**. The **Variable panel** opens and shows all the variables mapped in the data file.
2. Double-click the variable we want to find.

Design Manager takes us to the first data area mapped with that variable. We can search for the next occurrence of the variable by right-clicking in the Edit Panel and selecting **Find > Selected Variable**. Select **Next Variable** to find the next mapped variable.

4. Navigating by Customer or Section

We can move quickly through a data file object by jumping from one customer or section to the next.

Customer:
Right-click in the Edit Panel and select **Find > Prev Customer**. Design Manager takes us to the previous customer. Right-click and select **Find > Next Customer** to go to the next customer.

Section:
Right-click in the Edit Panel and select **Find > Prev Section**. Design Manager takes us to the previous section. Right-click and select **Find > Next Section** to go to the next section.

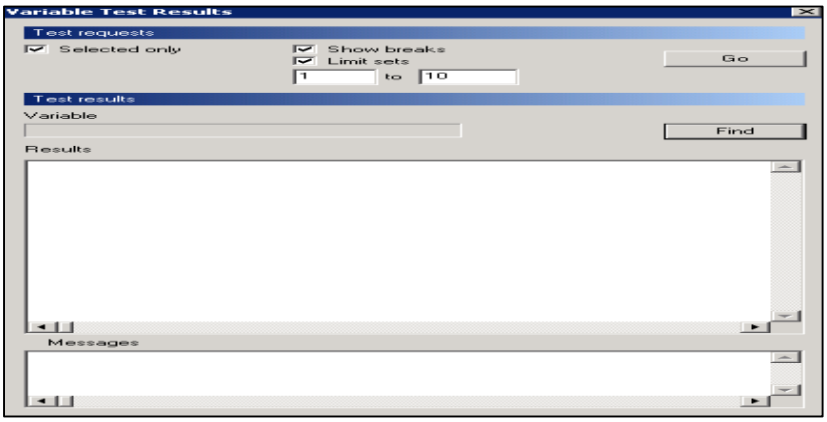
Testing Variable Mapping

We can test a selected variable or all the variables in a data file object to make sure variables are mapped correctly. The test shows the corresponding information in the data file object for one or several customers.

To test variable mapping:

1. Click a mapped data area in the Edit Panel to highlight it.
2. Right-click and select **Test**.

The **Variable Test Results** dialog box opens.



Test Requests Area:

The **Test requests** area lets us specify what variables are tested.
Options are:

- **Selected only**—Displays the value of only the variable mapped to the highlighted data area in the Results box. Clear this check box to test all mapped variables in the data file.
- **Show breaks**—Displays where each customer ends in the **Results** box.
- **Limit sets**—Displays only a sample of the data file in the **Results** box. Enter a beginning and ending range in the boxes.

Test Results Area:

When we click **Go**, Design Manager displays variables and their mappings in the **Results** box. If mapped incorrectly, error messages appear in the Messages box. If we choose **Selected only** in the Test requests area, the variable name appears in the Variable box.

2.2.11 COBOL Copybooks

HP Exstream supports a COBOL copybook feature to reduce the time spent on creating and mapping variables in a columnar file.

COBOL copybook lets us import a complete record layout into an unmapped columnar data file object and select the variables to be imported. Design Manager will then create the variables as is specified in the copybook file and map them to the corresponding data areas. We can also use the already existing variables (in the Library) as well.

NOTE: *COBOL Copybook auto-map is not supported in the DBCS version of HP Exstream. Also, HP Exstream supports COBOL constructs, including COMP.*

Benefits

It significantly reduces the time needed to create variables and map data areas.
Beneficial specially when the data file contains large number of fields and mapping manually would be hectic. Also when we want to add a field or change an existing mapping in the middle of a tag structure in a data file, using copybooks would be useful.

2.2.11.1 COBOL COPYBOOK SAMPLE

Following is a sample copybook for mapping three tags-

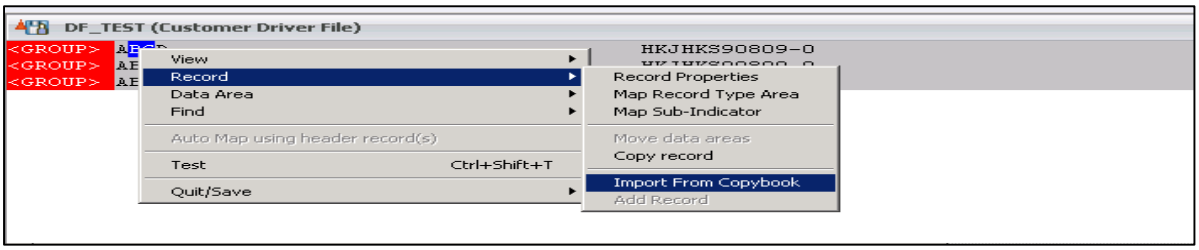
<HEADER> → with 3 fields
<GROUP> → with 6 fields
<TRAILER> → with 3 fields.

```
01  HEADER
    05  HEADER-DATE            PIC X(10).
    05  HEADER-TIME            PIC X(10).
    05  HEADER-DETAIL          PIC X(50).
01  GROUP.
    05  GROUP_NAME             PIC X(50).
    05  GROUP_ID               PIC 9(9).
    05  GROUP_CUMB-ID          PIC X(10).
    05  GROUP_ADDR             PIC X(50).
    05  GROUP_STATE            PIC X(20).
    05  GROUP_CITY             PIC X(10).
01  TRAILER
    05  GROUP_COUNT            PIC X(10).
    05  CHECK-COUNT            PIC X(10).
    05  CHECK-AMT              PIC X(10).
```

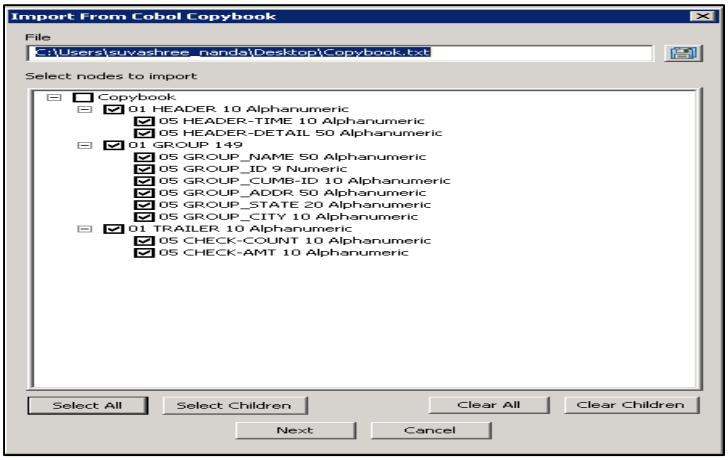
2.2.11.2 USING COBOL COPYBOOK FILE

We must first create a copybook structure before importing it so that it can be used to map variables.
To start the copybook process:

1. An unmapped data file is dragged to the **Edit** Panel.
2. On any data area right-click and from the shortcut menu, select **Record** > **Import From Copybook**. The **Import From Cobol Copybook** dialog box opens.



3. In the **File** box, the name of the external copybook file that is to be used is entered or browsed. The contents of the file appear as a tree structure in the **Select nodes to import** area.



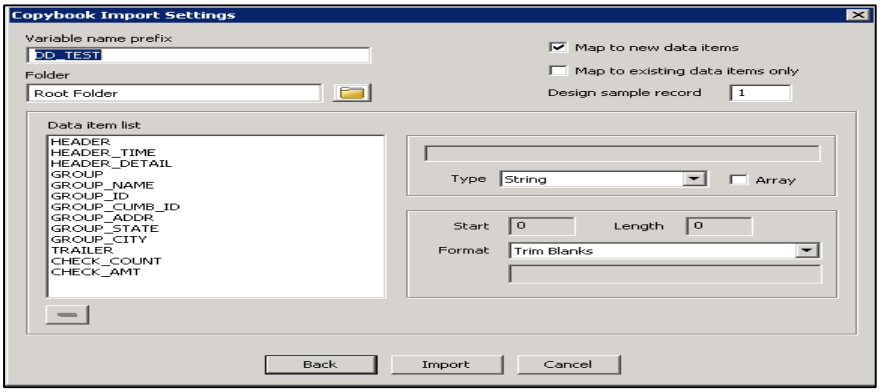
2.2.11.3 Specifying Variables for Copybook Import

After the copybook file is selected, the variables that will be imported to the data file object is specified.

1. We will select the checkbox to specify the nodes (variables) needed to map the current record. To speed this process, we can click the **Select All** or the **Select Children button** to select multiple items at once. To clear the options, we can click the **Clear All** or **Clear Children button**.
2. Click **Next**.
   The **Copybook Import Settings** dialog box opens.

2.2.11.4 Defining Variables to Copybook Import

The **Copybook Import Settings** dialog box allows us to define the imported variables.



The various fields here are

- **Variable name prefix:** To specify the prefix to the new variables that will be created so as to follow an application-specific naming convention.
- **Folder:** To specify the folder where the new variables will reside in the library.
- **Mapping Method:**
  To specify the mapping of variables any one of the following checkboxes can be selected.

- **Map to new data items**: This option will create new variables. In the **Design sample record** box we specify the data area that will be used to populate the design sample for the new variables.
- **Map to existing data items only**: This option will use previously created variables, either manually or in a previous Copybook execution.

NOTE*: To create the variables but generate no data mappings, we will need to clear both boxes as specified above.*

- **Data Item List:** To define the format of the variables this option is used.
  - A variable is selected. Its corresponding **Type** and **Format** is defined.
  - For a variable that can have multiple values **Array** checkbox is selected.

Click **Import**.

Design Manager maps the data file object with the chosen variables in the sequence they appear in the list. The lengths of the variables are the same as those in the original copybook file.

NOTE: *To adjust the positions we can right-click the data area and access its **Data Area Properties** dialog box.*
*We cannot adjust the **Start** or **Length** settings in the **Copybook Import Settings** dialog box.*

2.3 OUTPUT QUEUES

2.3.1 Introduction to Output Queue

- Output queue is a channel which is required to deliver the composed output to different customers.
- An application can have multiple output queues, including the mix of Print and Electronic media.
- Rules are applied on output queues to specify different queues for different documents, as per the requirement.

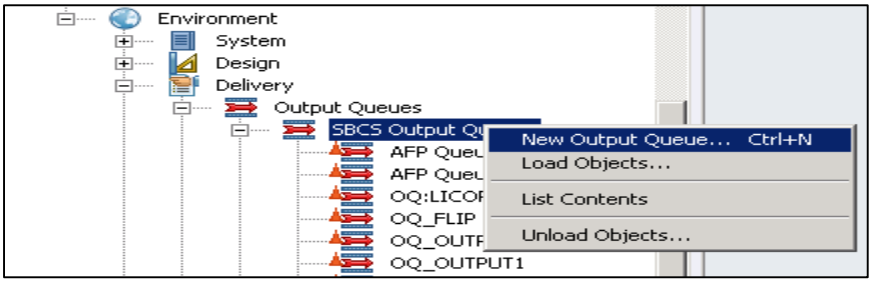
Uses of Output Queue

- To control multiple devices and process in final output production.
- Create multiple output files in a production run.
- Send output to different destinations, such as to printers (using pDrivers) or in electronic form (using eDrivers).

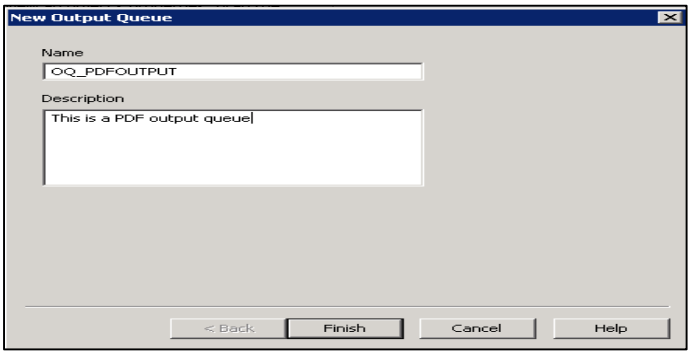
2.3.2 Creating an Output Queue

To create an Output Queue:

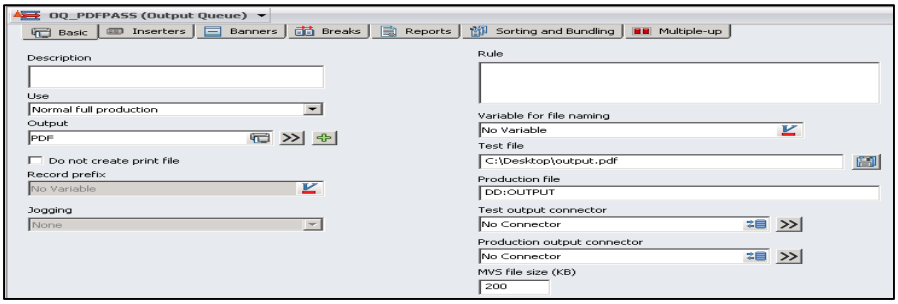
1. Right click the **Output Queue** heading from **Environment** -> **Delivery** -> **Output Queue**, select **New Output Queue**.



2. Enter a name in the **Name** box and description in the **Description** box (optional).
3. Click **Finish**.



4. The Output Queue appears in the Library and opens in the Property Panel to define.



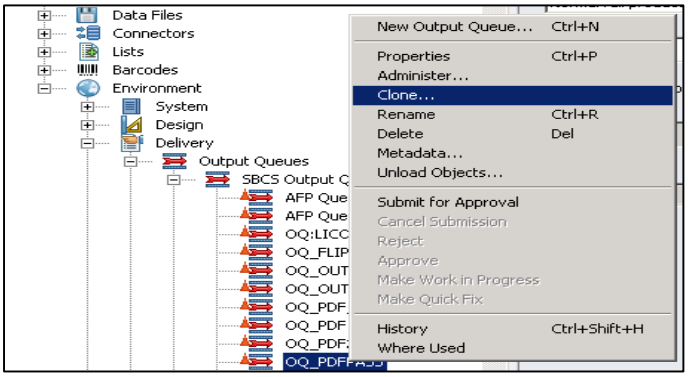
NOTE: *Before creating an Output Queue, at least one output object needs to be defined.*

2.3.3 Cloning an Existing Output Queue

- It is not mandatory to create a new output queue for each application; the same existing queue can be used by cloning the existing queue object.

To clone an existing output Queue:

1. Right click on existing **Output Queue**, select **Clone**.
   The Clone Output Queue dialog box opens.

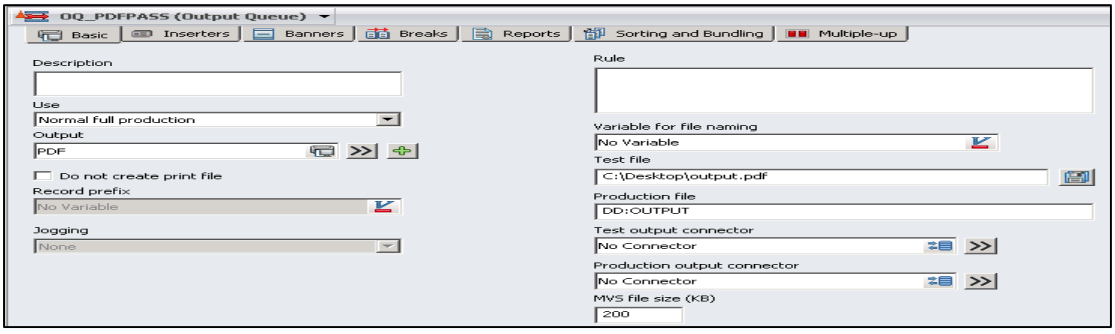


2. Enter the **Name** and **Description** (optional) in the box.
3. Click **OK**.

The Output Queue appears in Library and opens in the property panel for the definition.

2.3.4 Defining an Output Queue

- **Basic Tab**



- **Description:** Enter the description (Optional) which will differentiate one output queue from other in multiple applications.
- **Use:** Determines the production mode for using the output queue.
  The drop-down includes-
    1. **Normal full production:** For standard production use. (Default setting)
    2. **For postsort processing only:** Use with Output Sorting and Bundling module only.
- **Output:** Outputs are the objects which are used by queue to send to the customers. This can be in any of the printed format e.g. AFP output or can be viewed by customers directly in online or can be sent to customers in mail e.g. PDF output.
  Select the Output object from the **Output** box, as per the requirement for the application.
  Click  and select an Output object from **Select Output** dialog box.
  Click  to access and edit the output device properties.

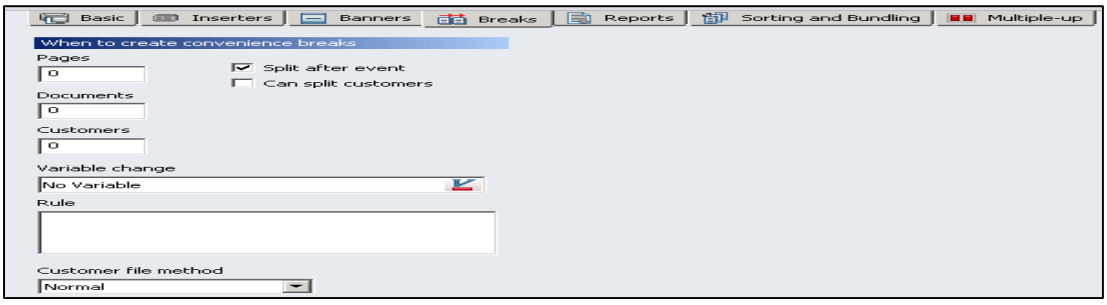
  The details about Outputs are discussed in this chapter at last.
- **Rule:** Rules are applied to the output queues to split output across different queues based on criteria such as number of pages or weight of documents or type of document etc.

  How to specify a rule in rule dialog box is discussed in RULE chapter.
- **Variable for File Naming:**
    1. A variable is selected to control the naming of generated output files by the output queue.
    2. Specifying a variable, overrides anything written in the **Test file** box.
    3. If a connector object is entered in **Production output connector** box or **Test output connector** box, the Variable for file naming box is inactive.
    4. If a variable is not selected and output queue has breaks, files names are determined by appending a numeric values to the end of the name specified in the **Test file** or **Production file** box.
       Example- If the file name is entered as C:\Output\Test.pdf then the first generated file name will be Test.pdf and second file name as Test2.pdf.
- **Test File:** A local file name is specified in the **Test File** box.
  By default, the engine writes the generated output to the HP Exstream program files folder. To store the file in a particular location, click 
- **Production File:** It is mandatory to specify a production file name in **Production file** box. For example- If the production environment is z/OS then the Production file name should be in upper case and begin with prefix DD: followed by max eight characters.

- **MVS File Size (KB):** If z/OS platform is used for production, the expected output size is specified in **Kilobytes** in the box. Choosing too small a size results in slower production speeds. The default, 200KB, is a moderate size.

- **Inserts Tab:** Inserter pages can be added to the output queue using the Inserters tab.
- **Banners Tab:** Banner pages are print at the start or end of production events. These pages can be static or contain current information about the run. Banner page objects must be created in the library before assigning to the output queue.
- **Breaks Tab:** Divides the output of an output queue in different output files. The breaks can be based on the following:
    1. Number of pages
    2. Number of documents
    3. Number of customers
    4. Variable change
    5. Rule

  The check box **Can split customers** is checked to permit documents of a customer to be split into multiple output files. And clear of this check box keeps all documents for a customer within a single output fie.



- **Reports Tab:** A custom report file is created by engine for various types of reports about the output queue.
  For Example- We can have the report generated for each customer processed or for each document sent to specific output queue.
- **Sorting and Bundling Tab:** The sorting and bundling of output files requires the **Output sorting** module.
- **Multiple-Up Tab:** Specifies the arrangement of customer pages on the sheet.

Output

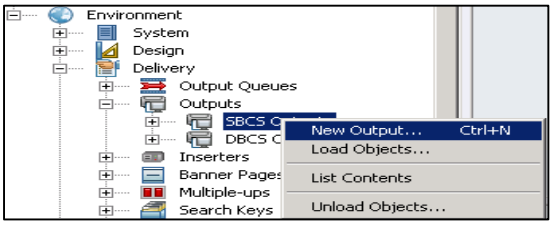
- An **Output** object defines the language and format the Engine uses to create the final output for a run.
- The specific options for an Output object vary widely depending on the selected output driver.
- Both P driver (for print media) and e driver (for electronic media) require output object.

2.3.5 Creation of Output object

All output objects resides under **Output** heading under **Delivery** in **Environment**.

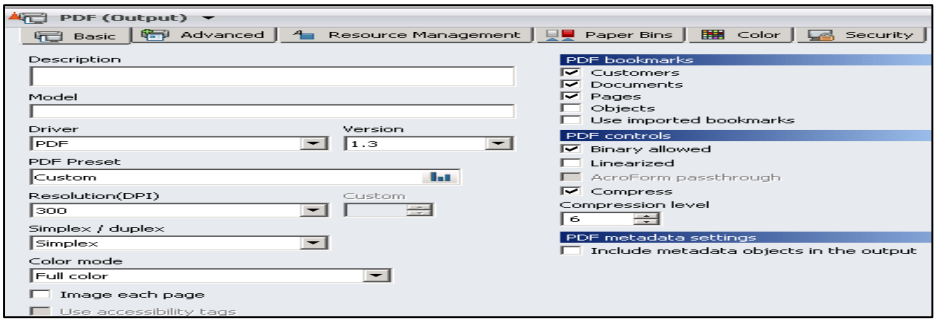
To create a new output object:

1. Right click **Outputs** heading.
2. Select **New Output** from the shortcut menu.

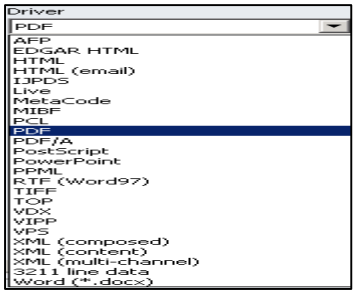


3. Enter a **Name** for the object and a **Description** (optional).
4. Click **Finish**.

The output object appears in the Library and opens in the property panel for defining.



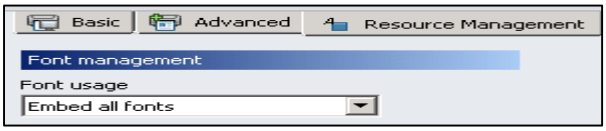
- **BASIC TAB:**
  - **Description:** The description is the short information about the output object entered while creating the output. This can be changed in property panel of output.
  - **Driver:** The tabs, properties and options available in output properties vary significantly with the selection of particular Driver from the various driver options.



  - **Resolution:** Resolution is specified for the destination output or display device. Again this varies with the selection of output driver. The measurement of resolution is in **dpi** (dots per inch).
    For Example: Here we will mention some of the driver along with few resolutions
    AFP/PDF/Post Script- 72, 240, 300, 480, 600, 1200, 1440, 2400
    PCL- 300, 600, 1200

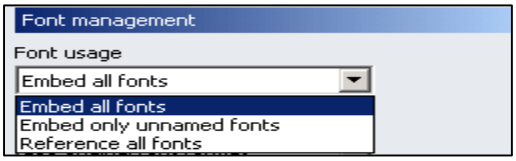
- **Simplex/Duplex:** The option has Simplex, Duplex, or Simplex and Duplex.
  The Engine use Document and page properties to determine which page to print as simplex or duplex.
  System variables that count sheets correctly take into account which pages are simplex and which pages are duplex.

➢ **RESOURCE MANAGEMENT TAB:** This is used to manage the output resources that affect the output file size and engine processing time by determining how the output handles fonts, images, and overlays.
Here we will be discussing about Font Management only.



FONT MANAGEMENT

1) **Embeded all fonts:** Includes a copy of all fonts used in the application in print file. This setting results in larger file size than other options but it does not affect the processing time and provides the most control over the printed output.
2) **Embeded only unnamed fonts:** Includes only those fonts that are not defined in Exstream.
3) **Reference all fonts:** Reference all fonts for the application in the print file. When this option is used, it is possible for the output to look slightly different from the design.



2.3.6 Different types of Output Driver

Here we will be discussing some of the output drivers only.

1) **AFP Output:**
   - The AFP output driver produces the necessary resources to support the print stream for IBM's Advanced Function Presentation (AFP) format printers.
   - AFP is a part of pDrivers.
   - The resolution is set to specify the sharpness of the text and graphics which is measured in dots per inch (dpi)
   - Search keys are used in AFP for tagging points in an application before or after each page, document, or customer.

   The details about search keys are discussed in SEARCH KEYS chapter.
2) **TIFF Output:**
   - The TIFF output driver produces Tagged Image Format File (TIFF) graphic file output.

- ❖ TIFF output is viewable on most graphic viewers and maintains a high degree of graphic quality.
- ❖ TIFF output converts each page in the design to a black-and-white bitmap file, compressed with TIFF Group 4 compression, the standard format for faxing or for storing documents in archival and retrieval systems.

3) **PDF Output:**
   - ❖ PDF output objects have unique properties that allow controlling how bookmarks are included in the output.
   - ❖ PDF is a part of eDrivers.
   - ❖ PDF outputs are used for viewing over the web, for delivery to customers through email, for standard printing, for archival storage and also in PDF output hyperlinks can be added.
4) **RTF Output:**
   - ❖ The RTF output driver produces Rich Text Format (RTF) files that are ready for viewing and editing in word processing programs.
   - ❖ These files are used most often for communications that customers can revise in word processing programs such as Microsoft Word.
   - ❖ HP Exstream supports RTF output for Microsoft Word 97 and later.
5) **3211 Line Data:**
   - ❖ The 3211 line data output driver produces the necessary resources to support the print stream for 3211 impact line printers.
   - ❖ HP Exstream supports many features of 3211 line data output, including record separation, text overflow, ANSI or machine carriage controls, ASCII or EBCDIC output character sets, number of lines per inch, number of characters per inch, blank line removal, and line overprinting.
   - ❖ Line printers produce output with one font, printing only from the top to the bottom of the page and one line at a time. This technology is frequently used for high-volume printing.
6) **XML Composed:**
   - ❖ The XML (composed) output driver produces fully-composed XML files for the Exstream Exchange Format (DXF).
   - ❖ The XML files we create with the XML (composed) output driver include the entire page layout, formatting, and all objects sent to the print stream (for example, images and pages). These files are ready for Web presentation.
   - ❖ The output contains all the objects sent to the print stream (including shapes, images, pages, and so on). This makes the output file very large.
   - ❖ XML (composed) output supports the inclusion of hyperlinks in the output file.

7) XML Content:

- The XML (content) output driver creates XML output that documents the logical content of the design. Logical content describes the structure and textual content of a document, but not the layout or formatting (for example, color and font).
- XML (content) is most often used for archival and internal statement regeneration programs.

8) Post Script:

- The PostScript output driver produces the necessary resources to support the print stream PostScript printers.
- PostScript output automatically builds forms and images and places them at the top of the print stream so they can be pre-ripped and referenced. PostScript output supports full-color, highlight color, grayscale, and black-and-white color modes; and paper bin selection.
- PostScript is a machine-independent output driver that supports macros and commands. Its user-friendly codes control formatting options such as page layout and scale outline fonts.
- PostScript offers font rendering and graphics that remain faithful to the screen.

2.4 VARIABLES

2.4.1 Introduction to Variables

- A variable is an object in the Design Manager Library that represents data that changes at engine run time from sources such as customer data, current date and time, pages in a document, or the value of other variables passed from upstream in the data file.
- Variables personalize the communications.
  For example, if we are creating a letter, we may want to include the customer's name in the letter text. We can create and insert a variable which inserts the customer's name when the letter is produced as output.

2.4.2 Types of Variables

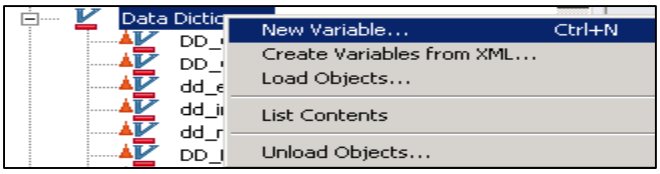
- **System Variables –** Variables that are predefined in HP Exstream. These are included with the HP Exstream product and HP Exstream automatically sets the values of these variables. System variables cannot be renamed or deleted. These are named with a prefix of 'SYS_'.
- **User Defined Variables -** Variables that are created, defined, and made available for design use by an administrator. These are named as per the naming convention used by user or administrator.

Benefits
Variables are key components for personalizing output since the value of variable changes dynamically.
Uses
Variables can be used to personalize any content, to track changes, create reports, write rules and hold a place in design to import external content.

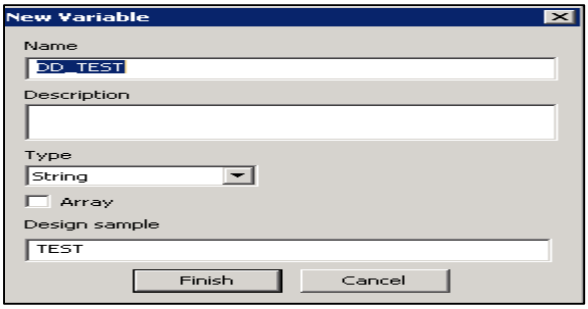
2.4.3 Creating a Variable

To create a string variable:

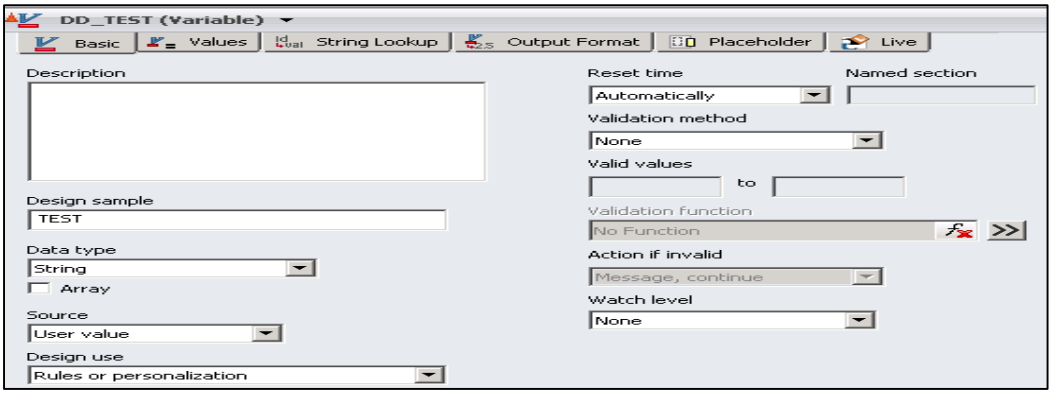
1. In the Library, right-click **Data Dictionary**, select **New Variable**.



2. In the **Name** box, enter a name.
3. In the **Description** box, enter a description (optional).
4. From the **Type** drop-down list, select the type of variable.
5. Select the **Array** check box if the variable contains more than one value.
6. Enter the default sample text in the **Design sample** box (optional). This text appears in Designer to represent the variable.
7. Click **Finish**.

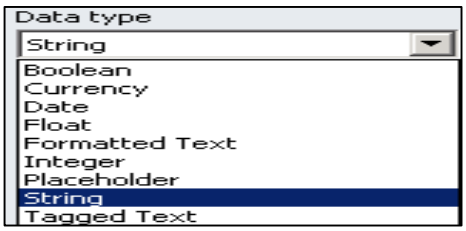


The string variable opens in the Property Panel for us to define.



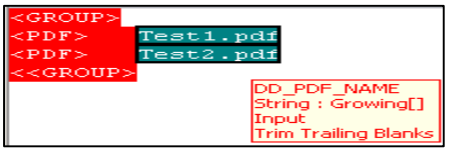
2.4.4 Defining a Variable

- **Basic Tab:** This tab is used to define the basic properties of the variable. The **Basic** tab contains properties that are common to all variables.
  - **Description:** Text entered in the description box is optional.
  - **Design Sample:** A text string is entered which will appear when the variable is inserted in any object in the Designer.
    For example, if we have a variable DD_CUSTOMER_NAME" created in the Design Manager and we want to use it in the page then if we have mentioned the design sample for the variable as "John" then instead of the entire variable name "John" will appear in the page when the variable is mapped inside the object.
  - **Data Type: Data type** drop-down list can be used to change the variable type. For example, if we have created a string variable but then decided to make it a Boolean variable then we can use the **Data type** drop-down list to change the variable type.

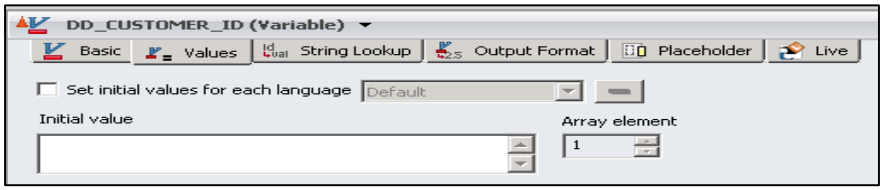


TYPE	DATA REFERENCED
Boolean	True or false
Currency	Currency Monetary amount
Date	Date (with or without time of day)
Float	Numeric value with decimals.
Formatted Text	Formatted content from an external repository or file source
Integer	Whole number
Placeholder	Content from an external text or image file during an engine run. This type of variable requires the Dynamic Content Import module.
String	Text
Tagged Text	Tags specifying the formatting of imported text

- **Array:** The **Array** check box is selected if the variable has multiple values for the customer or section of data.
  For Example: If we have a customer having two different values for a single variable then we have to mention it as array. We have to only map single variable in the data file then all the values coming under same tag will be automatically mapped with that variable.



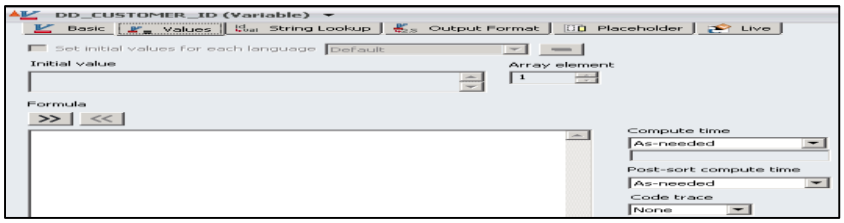
- **Source:** This option is used to define the method for generating the values of the variables. The options are-
  1. **User value:** The value(s) are entered on the Values tab in the Initial Value box.



  For example- Let us take a user variable (DD_CUSTOMER_ID) and set the initial value of this variable as "1" in the Initial value box. If we will use this in page then we can directly map this variable in the page or we can use this variable as an indicator for triggering any object inside page i.e. if we want to trigger a particular column of a table for once for 1st customer and for rest of the customer the column should exclude then we will use this variable to trigger for 1st time and then by using rule inside the table we will set the value of DD_CUSTOMER_ID as 0.

  2. **Formula:** The variable is computed using information from the database, based on the formula entered on the Values tab in the formula in the Formula box.

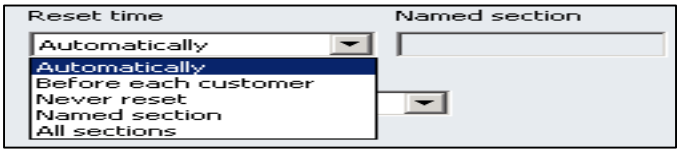
For Example- If we want to specify the location of any external file then formula is choose from the drop down to enter the path in the formula box.



3. **File only:** The value originates in a data file.
   These variables are used in data file and the same is used directly in pages or can be used in other formula variables for other manipulations to the value of the variable from file.
4. **Counter:** The variable increases by one during the run, according to a formula we have created on the Values tab.

❖ **Reset Time:**

Variables can be reset at specific times depending upon the requirement.
For example, if a variable populates account numbers for each customer, reset the variable before each customer to populate the correct account number for each customer.
The **Reset time** drop-down list is active if we select **User value**, **Formula**, **File only** in the **Source** drop-down list. From the **Reset time** dropdown list, select one of the following options to define when we want the variable to reset:



1. **Automatically**: The variable resets before each customer. If the variable is mapped in a section-based data file, it is also reset before any section is read. This is the **default**.
2. **Before each customer**: The variable resets only before each customer is read, regardless of section-based data.
3. **Never reset**: The variable resets only at the beginning of each engine run. Do not use **Never reset** for variables that represent a unique value for each customer.
4. **Named section**: The variable resets before a named section in the data file. This lets us create subtotals before a named section in the data file. In the adjacent box, enter the section name exactly as it is specified in the data file.
5. **All sections**: The variable resets before all sections.

➢ **Values Tab:** The properties of this tab is defined when Constant user value, Formula, or Counter is selected from the Source drop-down menu.

❖ **Initial Value:** The box is active if **User value**, **File only**, or **Counter** is selected from the **Source** drop-down menu.

In the **Initial value** box, an initial value can be specified up to 64K characters.

- **Formula Area:** In the **Formula** area, enter logic, such as the computation of a value or a counter. Formula logic uses generally accepted programming practices modeled after Visual Basic.

  All formulas must begin with one of the following prefixes:
  Value =
  Integer =
  [Variable name] =

  Function always returns a single value so it is mandatory to include one of the prefixes before the logic.

  If a formula variable has the **Array** check box selected in the **Basic** tab, then we must add (i). The subscript "i" acts as a counter.

  For example:
  Integer i
  Value(i) = variable1 + variable2
  NEXT i

- **Placeholder Tab:** The **Placeholder** tab contains all options specific to placeholder variables. The options on this tab are active only if the **Data type** is specified as **Placeholder** on the **Basic** tab.

2.4.5 Brief Idea on Placeholder Object

What is placeholder?
An object used to reserve an area where external content will be imported at run time.

What is placeholder variable?
A variable that reserves a place for external content the engine imports during an engine run.

NOTE: *The Dynamic Content Import module is must for importing the external content into the document using placeholder variable at run time, the external file can be images, logos, text files etc.*

Benefits of Dynamic Import:

- Dynamically import of external content, located in some network, into the application.
- Frequent change in external content does not require the Engine to package every time.

Dynamic Content Import can be done in two ways:

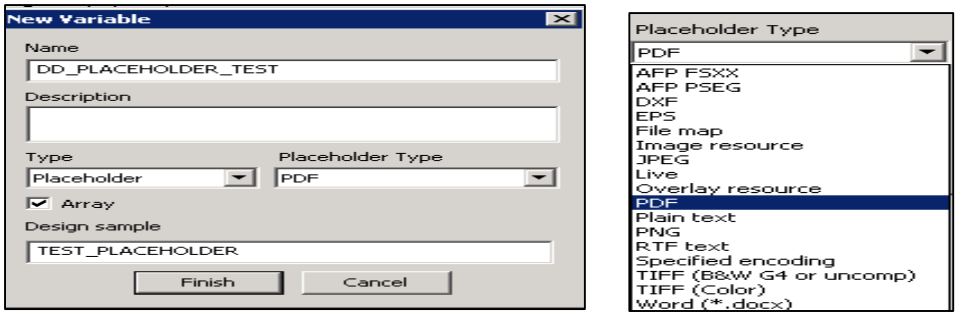
- **Dynamic import (Normal Compose):** Places imported files and images directly into the application and converts unsupported formats to supported ones.
- **Pass-through (Pre-compose):** References files and images without adding them to an application. It ignores files in unsupported formats, rather than converting them.

2.4.6 Creating Placeholder Variables

The Dynamic Content Import module is required to create and define placeholder variables. We use placeholder variables with the Dynamic Content Import module to hold a place in an object for external content to be imported when we run the engine.
For example, to dynamically import a TIFF file into a table, we create a placeholder variable and place it in the table cell.

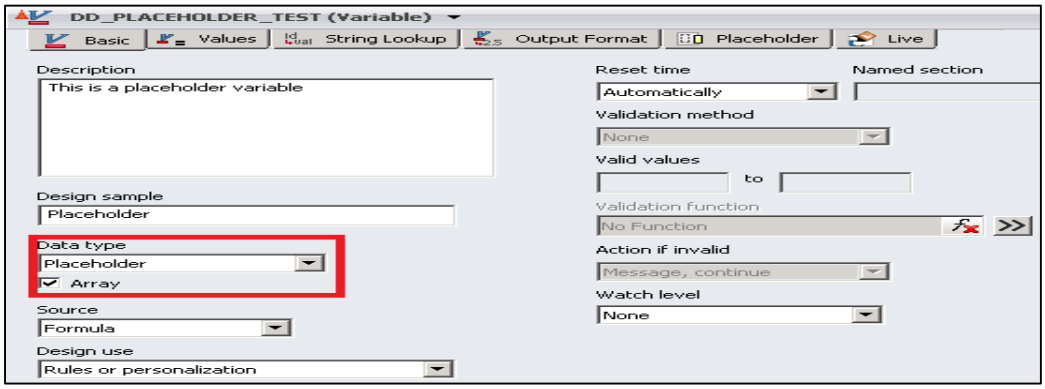
To create a placeholder variable:

1. Right-click the **Data Dictionary** heading in library and select **New Variable**.
   The **New Variable** dialog box opens.
2. In the **Name** box, enter a name.
3. In the **Description** box, enter a description (optional).
4. From the **Type** drop-down list, select **Placeholder**.
5. The **Placeholder Type** drop-down list appears.
   From the **Placeholder Type** drop-down list, select the type of file; we want the placeholder to dynamically import or pass through.

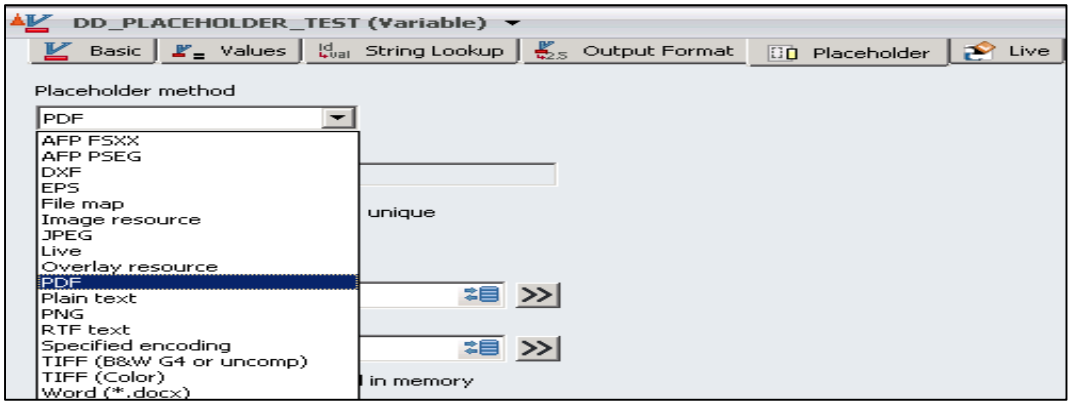


6. Select the **Array** check box if this variable has more than one array element (if it references more than one external file).
7. In the **Design sample** box, enter a design sample (optional).
8. Click **Finish**.

   The variable opens in the Property Panel to define.



The **Placeholder method**, in placeholder tab, drop-down list specifies the type of data being imported.

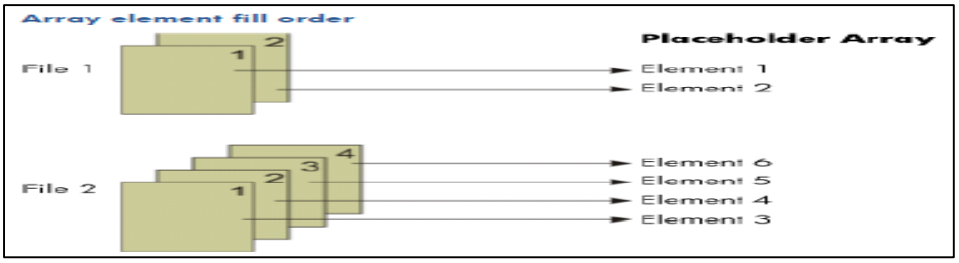


Type of File	Description
AFP FSXX	Provides pass-through of FS45, FS10, and FS11 IOCA images.
AFP PSEG	Provides pass-through of AFP page segment image files.
EPS	Provides pass-through of EPS images.
File Map	Provides pass-through of PDF XFA forms mined for data (with the PDF form miner module).
Image resource	Imports image files such as PostScript TIFF images or Metacode IMGs on a printer. Loading image files at run time helps make the production output file smaller.
JPEG	Provides pass-through of JPEG color images.
Live	Imports data contained in a DLF and uses it to generate a new application (with the DLF Input module).
Overlay resource	Imports Metacode forms (FRMs), AFP overlays, PostScript forms, and page segments.
PDF	Imports PDF content.
Plain Text	Imports unformatted text files.
PNG	Imports PNG black and white images.
RTF text	Imports RTF files.
Specified encoding	For DBCS only, imports files with the specified encoding.
TIFF (B&W G4 or uncomp	Imports images compressed with the Group 4 TIFF standard or 1-bit uncompressed black-and-white images.
TIFF (color)	Provides pass-through of TIFF images.
Word (*.docx)	Imports DOCX content

2.4.7 Multiple-Page Import Placeholders

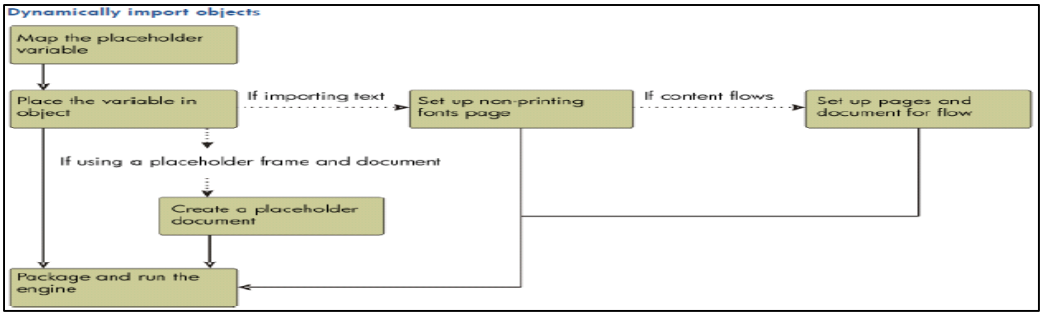
- A placeholder variable for multiple-page imports the same way we define the placeholder variables for other imports.
- Can import multiple-page TIFF, PDF, and DLF files into the document.
- Can import all the pages in the file, a range of pages, or pages specified by a variable.
- Can store large multiple-page files, such as images, outside the database and import them at run time.

- The **Array** check box is always selected to import multiple-page TIFF, PDF, or DLF files. Every page in a multiple page file becomes one element in the array.
- The array is loaded with the pages of each file after data is read but **before composition**.



2.4.8 Dynamic Import Object Flow

The flow of dynamically imports objects with either the Dynamic Content Import or the PDF Import as Image module.



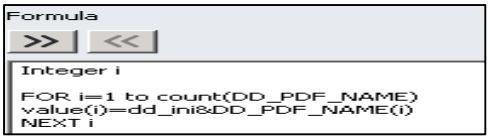
2.4.9 Dynamic Import Object for Normal Compose

As we have seen in previous section that the dynamic import can be done in two ways-

1. Pre-compose Placeholder (Pass through, already discussed in section 2.1.3.4 and 2.1.3.5) and
2. Normal compose

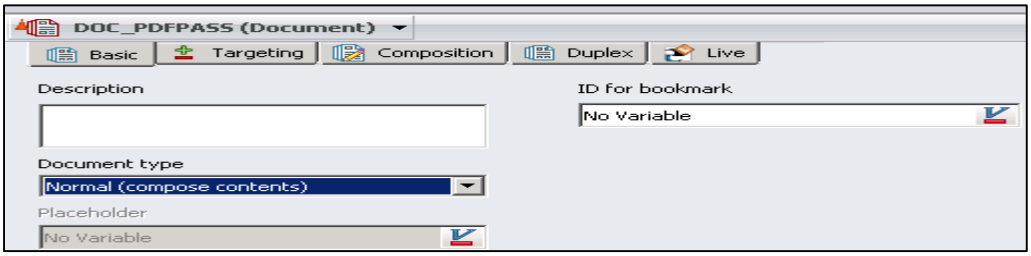
Steps for Normal Compose Dynamic Import-

1. Create a placeholder of required type i.e. TIFF, PDF etc.
2. Provide the path of the external file to be imported in the application in the values tab of placeholder variable. The path can be specified using initialization file or can be written directly in the Formula box of Values tab or can be specified the variable which contains the entire location of file from the data file.

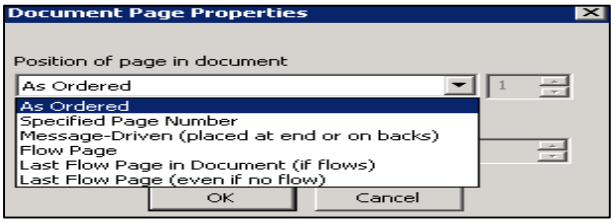


For Example- Here in the above snap shot the location of the external file is mentioned in a variable i.e. "**dd_ini**" which is mapped in the initialization file. And "**DD_PDF_NAME**" is an array variable mapped in the data file which holds the PDF name which required to import. Hence the value of the placeholder variable will be the PDF file having multiple pages.

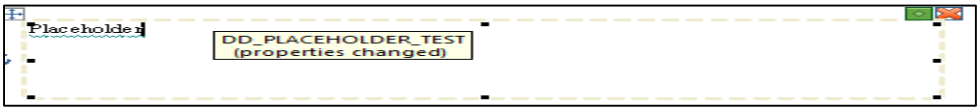
3. The document where the placeholder is used inside the page should be set as “Normal (compose content)” in the basic tab of the document.



4. The document page property (drag and drop the document to the edit panel) is by default set as “As ordered”.



5. The placeholder variable is inserted inside into a Text box or Table cell. Inserting the placeholder variable into the object helps in displaying the static text along with the imported object.



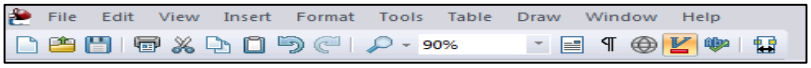
2.4.10 Variable Panel

The created variables can be inserted into the object or formulas or text using variable panel. The variable panel can be open using the following methods-

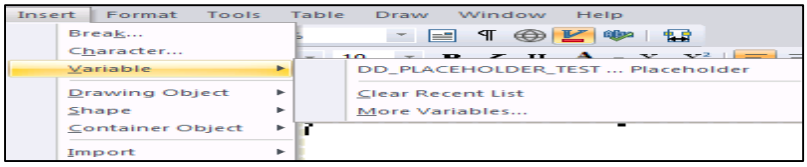
- **In Design Manager:**
  1. On the Toolbar, click  .

  2. Select **View > Variable Panel**.
  3. In the Edit Panel, right-click a data area and select **Data Area > Map Existing Data Item**.

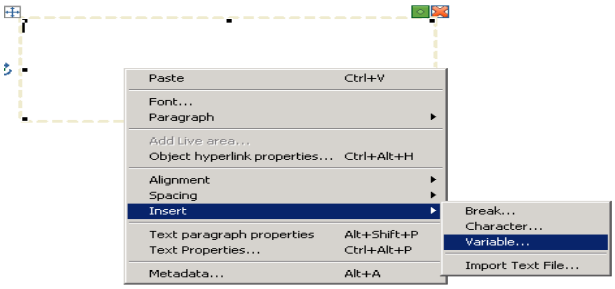
- **In Designer:**
  1. On the **Standard** toolbar, click 

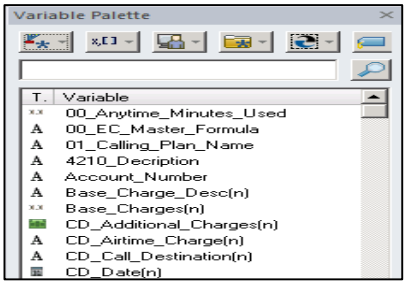


  2. From the **Insert** menu, select **Variable**.



3. Right-click the location where we want to insert the variable and from the shortcut menu select **Insert >Variable**. The variable palette with all the variables will appear in the right side.





2.4.11 Selecting a Variable

To add a variable to an object from the Variable panel, highlight the variable name and double-click it. We can also drag the variable from the Variable panel to:

- **Design Manager**: Either the Property Panel (such as to a property that has) or the Edit Panel (such as to a highlighted data area).
- **Designer**: An active object that accepts variables, such as a text box, or to a property in a dialog box that has a box with .

2.4.12 System Variables

HP Exstream provides a number of system variables which can be used to control devices. The following are some key aspects of using system variables:

- System variables are always kept in the **Data Dictionary** heading under the root folder.
- We can clone system variables.
- We cannot delete system variables.
- HP Exstream automatically adds four system variables to the package file, whether we use them or not:

❖ SYS_LanguageCustomer	❖ SYS_LocaleCustomer
❖ SYS_CustomerEffectiveDate	❖ SYS_CustInvalidDataLevel

Some of the System variables are as follows:

System Variable	Description
SYS_CustomerDocuments	The number of documents for the current customer.
SYS_CustomerInRun	The current customer number being processed.
SYS_DateCurrent	The date the engine is being run to produce documents (today). This is the system date.
SYS_DocumentTotalInRun	The total number of customers that have been written to all output queues.
SYS_PageInDocument	The current number of counted pages within the document currently written.

2.5 RULES, FUNCTIONS AND SUBROUTINES

2.5.1 Introduction

- HP Exstream provides various conditional and logical constructs which define how the Exstream Engine manipulates data and places objects on the final output.
- The input customer data is used to personalize the applications based on the logic it needs to incorporate.

  For example, we might want a page to be triggered only when the customer belongs to a specific state such as Connecticut or California, or we might also need to perform some conditional / logical processing based on Customer's State code and Zip code to trigger some specific page.

  HP Exstream provides components such as Rules, Functions and Subroutines to effectively handle such scenarios.

2.5.2 Rules

Rules are used to include or exclude an object depending on customer information.

For Example, when a particular customer meets all conditions in the logic, we can set the rule to include a special message or specific document.

The logic in a rule contains one or more separate comparative statements to include or exclude an object depending on customer information. When a particular customer meets all conditions in the logic, we can set the rule to either include or exclude an object from the output for that customer.

There are two types of rules in the design environment:

- **Library Rules:**

  Library rules are ideal when we need to use the same logic in several places. If we create a rule and save it in the Library, we do not have to recreate the logic each time we need to use it. They are also called **Named Rules**.

  To create a Library rule:

  1. In the Library, select the Rules heading.
  2. Right-click and select New Rule. The New Rule dialog box opens.

  3. In the Name box, enter a name. In the Description box, enter a description (optional).
  4. Click Finish.

  The rule opens in the Property Panel for us to define.

➢ Unnamed Rules

Use unnamed rules when we need logic to control only a specific object. How we create the object, and therefore the rule, determines how we access rules.
In Design Manager, click the Targeting tab when the object is in the Property Panel. In Designer, click the Rule tab in the object's properties.

Rule Dialog Box

The Rule dialog box lets us easily select or create Library rules. When working with design objects such as documents, pages, paragraphs, and sections, we can access it from the Targeting tab (in Design Manager) or the Rule tab (in Designer).

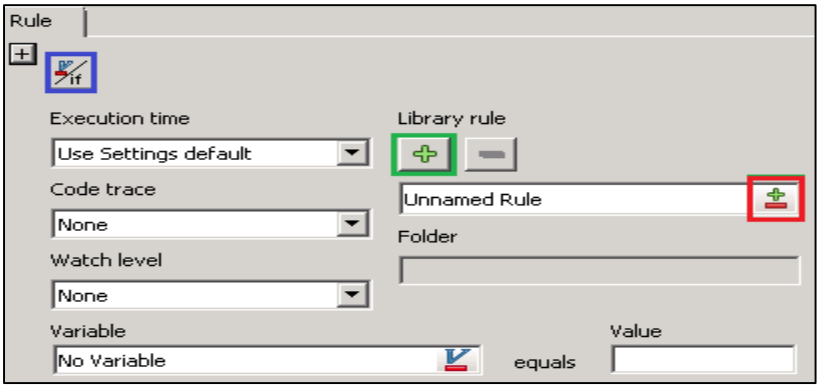
The Rule dialog box has three modes:

- **Equality rule mode**: Basic rules that select objects based on the singe value of a single variable.
- **Simple rule mode:** Simple inclusion or exclusion rules with no programming.
- **Code panel mode**: Complex rules that use a mix of AND and OR statement.

Rule dialog box (default view)

Rule dialog box (expanded view)

By clicking on the ⊞ button in the default view shown above, we get to see the expanded view of the Rule Dialog box. By entering values and making selections from drop-down lists and the Variable panel, we can assign Library rules to most objects.



➢ Execution Time

The Execute time drop-down list lets us specify the time, or times, when the rules are run. Select from the following options:

- **Use Settings default**: This is the default. Rules are run according to the timing specified in the Default variable substitution & rule execution time drop-down list on the edit settings.

- **Initial Engine run only**: Rules are run only during the first engine run when the interactive document is produced. Users cannot upload and use their own distribution lists.
- **Initial Engine & LiveEditor**: Rules are run during the first engine run and when users upload distribution lists in the LiveEditor. Users can preview their data in the LiveEditor.
- **Initial & final Engine & LiveEditor**: Rules are run during the first and last engine run and when users upload mailing lists in the LiveEditor. Users can preview their data in the LiveEditor. We can also use that data during the final production run, which usually produces output in a format other than DLF.

➢ **Code Trace**

The Code trace drop-down list lets us record information about each line of a rule as it is read by the engine. Options are:

- None: The rule code is not traced. This is the default option.
- Source Line: Each line of the rule writes to the debug file as it executes.
- Assignment: Each line of the rule writes to the debug file as it executes, along with the variables as they are used.
- All Variables: Each line of the rule writes to the debug file as it executes, along with the values of variables as they are used.

➢ **Watch Level**

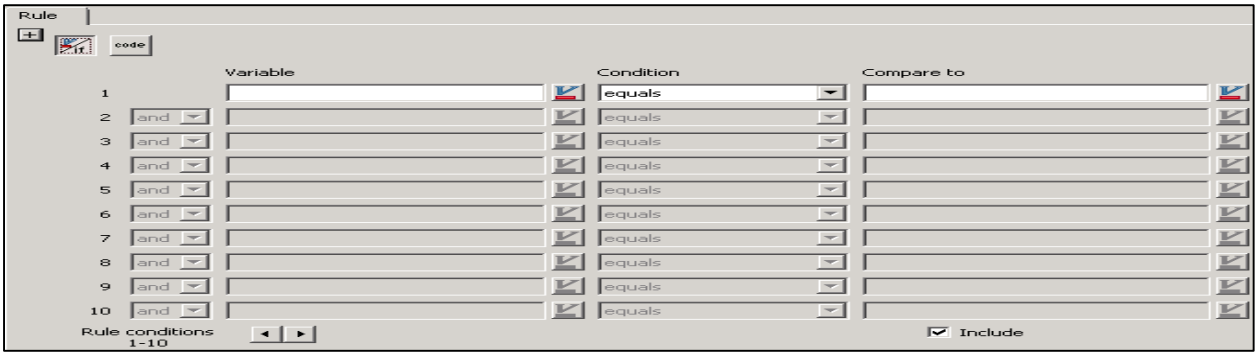
The Watch level drop-down list lets us determine if rules are included in a debug file. Select from the following options:

- None: Rule is not watched
- Fired: Engine adds information to the debug file each time the rule is run

While working in the Rule Dialog box, we can either create a new Library rule or we can also use an existing rule available in the library. We can the button (highlighted red above) to use an existing rule.

If we want to create a new rule we can use the button (highlighted green above). This will open up Folder dialog box where we can select the folder in which the rule is to be saved. It also opens up another dialog box after that to provide a name for the rule and its description.

Given below is an expanded view of Rule Dialog Box where we can assign variables, their conditions and values to be compared with. We can reach to this dialog by clicking on button (highlighted blue above).



We can also create the above conditional logic by writing code and IF-ELSE constructs in them.

By clicking on the **code** button above, we can get to the code panel.
Variable and **Compare to** columns above are used to compare a variable value to another variable or value defined in 'compare to' column.

Condition
Select a Condition that compares our choice in the Variable column to our choice in the Compare To column. Options are:

- **Equal To -** Variable must match 'Compare To' exactly.
- **Not Equal To -** Variable can be anything other than 'Compare To'.
- **Greater Than -** Value of Variable must be more than 'Compare To'. This can be used alphabetically.
- **Greater Than or Equal To -** Value of Variable must be the value of 'Compare To' or more. This can be used alphabetically.
- **Less Than -** Value of Variable must be less than 'Compare To'. This can be used alphabetically.
- **Less Than or Equal To -** Value of Variable must be the value of 'Compare To' or less. This can be used alphabetically.
- **Is Like -** Variable must be similar to 'Compare To'.

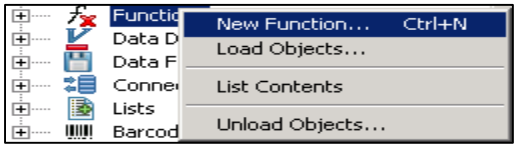
2.5.3 Functions

Functions are special pieces of code that cause an action to occur based on information in a customer's data. We can create our own functions in Design Manager using Visual Basic-style code.

2.5.3.1 Creating Functions

To create a function we will follow the below steps:

1. In the Library, right-click the **Functions** heading, and, from the shortcut menu, select New Function.

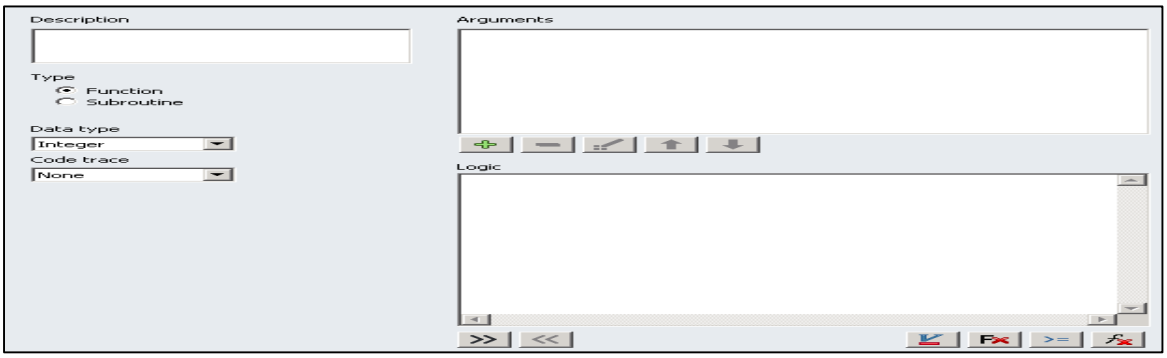


2. In the Name box, enter a name. In the Description box, enter a description (optional).
3. Click **Finish**.

NOTE: *We should not create a Library function with the same name as a variable. If we do, the variable is used instead of the function.*

2.5.3.2 Function Properties

We use the Property Panel to define the function's argument list and logic.



- **Type:** We specify the type of function we are creating by using this radio button. We can select from the following options:

  Function: Analyzes data and returns a value

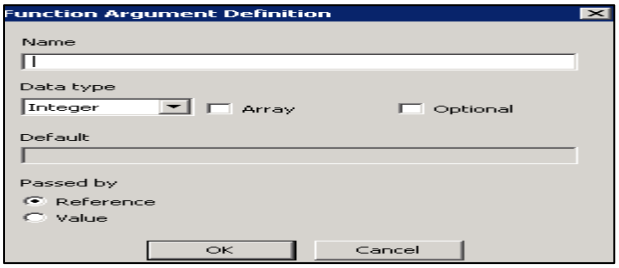
  Subroutine: Does not return a value. Instead, it performs an action based on data.

- **Data Type:** If we select the Function radio button, the Data Type drop-down list lets us specify the type of data the function returns. We can select from the following options:
  1. **String**: Text
  2. **Integer:** Whole number
  3. **Boolean:** True or false
  4. **Floating:** Numeric value with decimals
  5. **Date:** Date (with or without time of day)
  6. **Currency:** Monetary amount

- **Code Trace:** The **Code trace** drop-down list lets us record information about each line of a function as the engine reads it. We can select from the following options:
  1. **None**: The rule code is not traced. This is the default option.
  2. **Source Line**: Each line of the rule writes to the debug file as it executes.
  3. **Assignment**: Each line of the rule writes to the debug file as it executes, along with the variables as they are used.
  4. **All Variables**: Each line of the rule writes to the debug file as it executes, along with the values of variables as they are used.

- **Arguments Area:** We specify the arguments passed to the function or subroutine in the **Arguments** area.

  Adding Arguments: To create an argument, click  The Function Argument Definition dialog box opens.



Name: We must enter a name for the argument in the Name box. The argument is referenced in the function logic by this name.

Data Type: We can specify the below Data type for the arguments.

String: Text	**Integer**: Whole number
Boolean: True or false	**Floating**: Numeric value with decimals
Date: Date (with or without time of day)	**Currency**- Monetary amount

Array: Select the **Array** check box if the argument is an array variable.

NOTE: *If we select the **Array** check box, the **Passed by** options are inactive because an array is always passed by reference.*

Optional: Select the Optional check box if the argument is not mandatory for the function or subroutine.

NOTE: *Optional arguments must be inserted at the end of a function or subroutine's argument list. No mandatory arguments can come after an optional argument.*

Passed By: The Passed by radio buttons lets us determine how information pulled from the data is sent to the function. Options are:

1. **Reference**: Passes the variable and any modifications done within the function. These modifications are reflected when the value is passed to the variable outside the function. Arguments passed by Reference must be a variable, array, or array element, and the data types must match.
2. **Value**: Passes the value of the variable, not the variable itself. The function or subroutine does not affect the variable value outside the function.

Delete an Argument: To delete an argument, highlight the argument we want to delete and click 

Edit an Argument: If we need to make changes to an existing argument, highlight the argument and click  .The Function Argument Definition dialog box opens. Any changes we make are reflected in the Arguments box.

- **Logic Area:** We enter the logic for the function in the **Logic** area in two ways:
    1. Use an external text editor
    2. Enter the logic directly in the box using a combination of typing and selections from the dialog boxes opened by the shortcut buttons.

2.5.3.3 Using Functions and Subroutines

Once a function or subroutine is created, it is possible to call them from other functions, subroutines, formulas, and rules.

1. To call a function, precede the function with Value = or a warning is issued.
   Value = Function(argument list)
2. To call subroutines, we must precede the subroutine name with Call.
   Call Subroutine(argument list)

2.5.4 Inbuilt Functions

HP Exstream offers a number of built-in functions to make it faster and easier to perform routine data manipulation such as format conversion and value reporting. These built-in functions are available on the Select **Builtin Function** dialog box, which opens when we click  (rule Code Panel and the Library function properties) or select **Built-in Function** from the shortcut menu (Formula variable).

Some of the built-in functions available in HP Exstream are mentioned multiple times in this guide. This is because the functions are separated, using the same categories (except All, which lists everything) as the Filter drop-down list.

The following categories are:

- **Array:** Manipulate or return information about an array of data elements.
- **Date:** Provide information about dates.
- **General:** Perform a variety of actions. Most General functions are specific to the software.
- **I/O:** Set input/output (I/O) information for the engine.
- **Live:** Perform a variety of actions specific to LiveEditor.
- **Math:** Perform mathematical operations such as adding and averaging.
- **String:** Perform a variety of actions with string functions.

Some of the most commonly used built in functions are-

Builtin Functions	Type	Description
Contains	Array	The Contains function returns an integer that is the index of the first array element that matches Target. If there is no match of the Target within the array, we receive a 0 (zero) to indicate failure. **Syntax** Contains(Array, Target[, Start, Stop, Substring, NoCase, Sorted])
Count	Array	The Count function returns an integer that is the number of elements in a specified array. **Syntax** Count(Array)

Split	Array	The Split function splits a string into an array of substrings and returns an integer that is the number of Substrings created. **Syntax** Split(Array, String[, Delimiter, Limit])
Date	Date	The Date function returns the current system date. It can be used to specify the current date where necessary. **Syntax** Date()
LCase	String	The LCase function returns a string that has been converted to lowercase characters. It converts only uppercase letters. Lowercase letters and numeric characters remain unchanged. **Syntax** LCase(String)
UCase	String	The UCase function returns a string that has been converted to uppercase characters. It converts only lowercase characters. All uppercase and non-alphabetic characters remain unchanged. **Syntax** UCase(String)
Left	String	The Left function returns a string containing a specified number of characters from the left side of the string. **Syntax** Left(String, Count)
Right	String	The Right function returns a string containing a specified number of characters from the right side of a string. **Syntax** Right(String, Count)
Len	String	The Len function returns the length (integer) of a string expression. **Syntax** Len(String)
LTrim	String	The LTrim function returns a string without leading spaces. **Syntax** LTrim(String)
RTrim	String	The RTrim function returns a string without trailing spaces. **Syntax** RTrim(String)
Mid	String	The Mid function extracts a substring from a string. **Syntax** Mid(String, Start[, Count])
Replace	String	The Replace function replaces instances of a substring with another substring (for example, name or address changes). **Syntax** Replace(String, Find, Replace[, Start, Count])

2.6 SECTIONS AND PARAGRAPHS

2.6.1 Introduction to Sections

Sections are distinct parts or divisions of a document. It includes a single or a collection of **paragraphs** that can be used in multiple pages across same or multiple documents in an application.

Section objects are designed to hold paragraph objects or other section objects. The section and paragraphs objects can be arranged as per requirement.

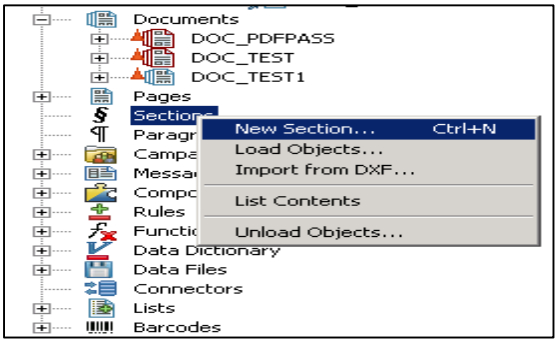
Benefits

- To group content and arrange the information in a high-level hierarchy.
- To create reusable and modular elements for documents.
- To create long flowing documents.

2.6.2 Creating Section Objects

Following are the Steps to create a **Section** object:

1. In the library, right-click the **Sections** heading, and from the shortcut menu, select **New Section**. The **New Section** dialog box opens.

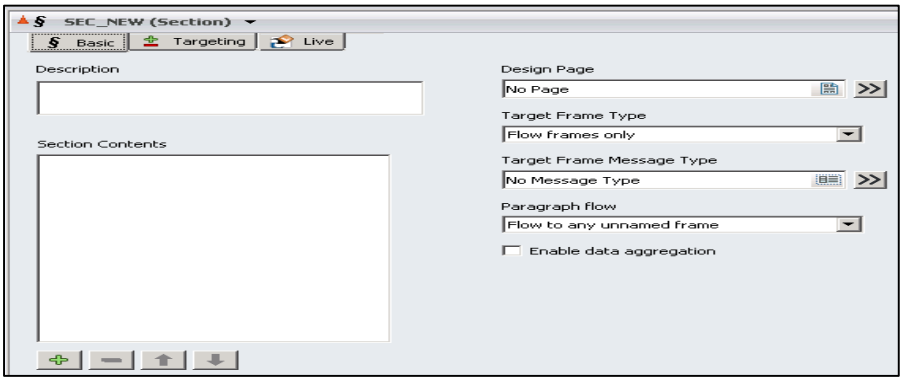


2. In the **Name** box, enter a name. In the **Description** box, enter a description (optional).
3. Click **Finish**.

The **Section** object opens in the Property panel.

2.6.3 Various Tabs in Section Object

- **Basic Tab:**



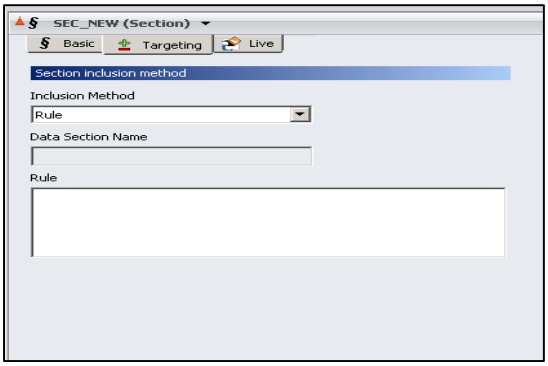
The Basic tab helps in defining the general properties. Following are the various fields in the basic tab:

- **Description**: This field is used to add description of the purpose of the section.
- **Section Contents**: This box is used to view the paragraph objects and section objects within the current section object. To further add/remove section and paragraph objects from the list click on  and  **.** To rearrange the contents sequence click on  and  .
- **Design Page:** This box basically contains the name of the page on which the section object is placed.
- **Target Frame Type:** This drop-down list lets us place section objects in specific types of frames. The following options are available:
    1. **Flow frames only**
    2. **Content frames only**
    3. **Content and flow frames**
- **Target Frame Message Type** → If we select **Content frames** only or **Content and flow frames**, the Target Frame Message Type drop-down list becomes active. The Target Frame Message Type drop-down list allows us to give paragraph objects within that section object a message type for use by the target frames. Also we must select the same message type for the frame as well.
- **Paragraph flow** → This drop-down list is used to control the flow of the contents of the section objects. In case the section object contains another section objects then the parent section object's paragraph flow setting is used. Following are the various options available under Paragraph flow:
    1. **Flow to any frame** →The contents of this section object can flow to any frame.
    2. **Flow to any unnamed frame** →The contents of this section object can flow to any unnamed frame (flow frame that doesn't have a flow name).
    3. **Flow to any named frame** →The contents of this section object can flow to any flow frame that has a flow name.
    4. **Flow to specified target** →The contents of this section object can flow to frames with a particular flow name only. The flow name is specified in the drop-down list that appears when this option is selected.

NOTE: *If a flow frame is present on the design page, the section object uses the width and column settings of the frame.*
If no flow frame is present, the section object uses the width of the page.

➢ **Targeting Tab**

This tab is used to specify the method the engine uses to select specific section objects for a personalized communication.



- **Inclusion Method**: This drop-down list allows us to specify how the section objects are selected. Following are its options:
    1. **Rule**—The section object is included when the document is composed, based on rules set in the Targeting Rule box.
    2. **Named Section**—The section object is included when the corresponding data section name is encountered in the data file. The name of the data-section associated with this section object is mentioned in the **Data Section Name** box.
- **Targeting Rule:** To specify the rules this field is used. It lets us select variables and conditions on which to base the rule.

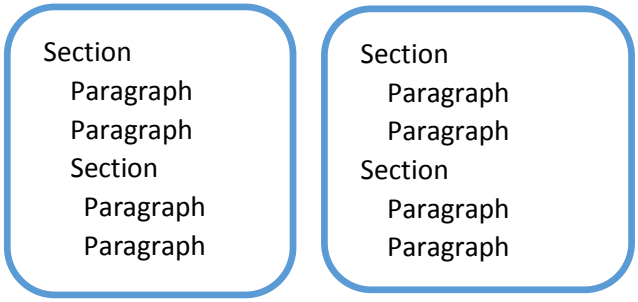
➢ **Live Tab**

The live tab is available only if there is license to the interactive document capability of HP Exstream (Live). It helps to determine how end users interact with the page when it is included in an interactive document.

2.6.4 Introduction to Paragraphs

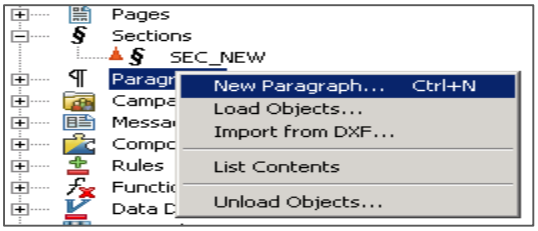
A paragraph is a group of sentences dealing with a single theme having sentences that support the main idea of that paragraph.

Section and Paragraph Object Examples:



2.6.5 Creation of Paragraph Objects

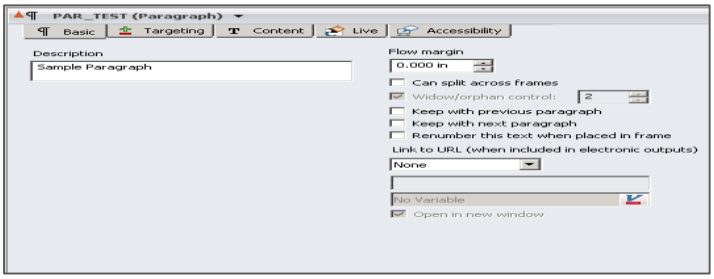
1. In the library, right-click the **Paragraph** heading, and from the shortcut menu, select **New Paragraph**. The **New Paragraph** dialog box opens.



2. In the **Name** box, enter a name. In the **Description** box, enter a description (optional).
3. Click **Finish**.
   The Paragraph object opens in the Property panel.

2.6.6 Tabs of Paragraph Objects:

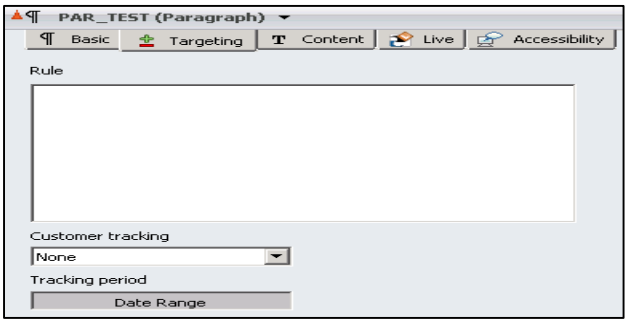
- **Basic Tab**



Following are the important properties of Basic tab:

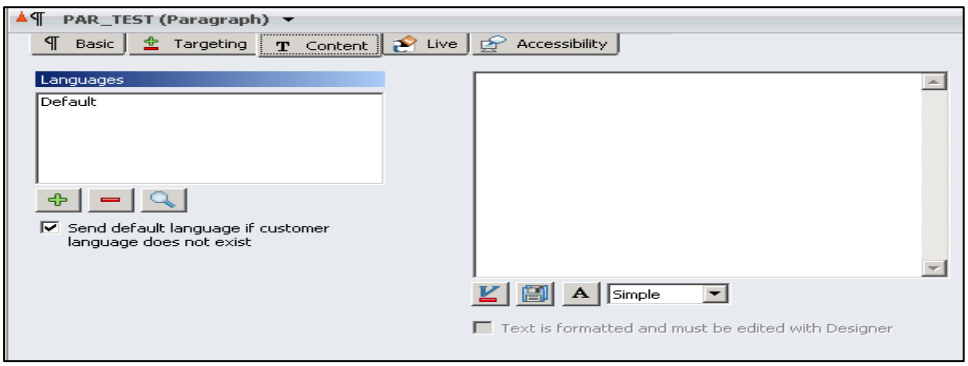
- **Flow Margin:** This option lets us set the minimum allowable margin between paragraph objects in the same frame.
- **Can split across frames:** This check box allows paragraph objects to begin on one frame and end on another (split) if there is not enough space to fit the entire paragraph object. When this option is checked, a suitable sized frame needs to be made available; else the paragraph object will not appear in the final output.
- **Widow/Orphan Control:** It is active only when **Can split across frames** is checked. It allows setting the number of lines of text that must be kept together when the paragraph object splits.
- **Keep with Previous Paragraph:** This check box forces a paragraph object to remain on the same page as the previous paragraph object.
- **Keep with Next Paragraph:** This check box forces a paragraph object to remain on the same page as the next paragraph object.

- **Targeting Tab**



This tab allows us to implement rules on paragraph object and track the customers who receive the paragraph object.

- **Content Tab**



The **Content** tab lets us view and edit the text of the paragraph object and any language layers.

- **Languages Area:** The Languages area controls language layers so the paragraph object can be distributed in the customer's native language.
- **Text Display Area**: The **text display area**, to the right of the **Languages** area, lets us view the content of a paragraph object.

- **Live Tab**

  The Live tab is available only if we have licensed the interactive document capabilities of HP Exstream (Live).

- **Accessibility Tab**

  The Accessibility tab is available if we have licensed the PDF module.

2.6.7 Putting Sections and Paragraphs on Page

Section objects and paragraph objects are placed in the design using **flow frames** and **content frames**. The frame to be used depends on when we want the section objects to be placed in the document. To associate a frame with a section, we first create a frame in the design page and place the created section in the same document as that of the page.

2.7 SEARCH KEYS

2.7.1 Introduction to Search Keys

Search Keys are **non-printing** records that are added to objects that can be read by the system, output device, or production equipment.

Benefits

- A search key is a method for tagging points in the print stream. These search keys can later be searched as proof that the document was sent.
- They help in identifying a particular customer based on the values retrieved.

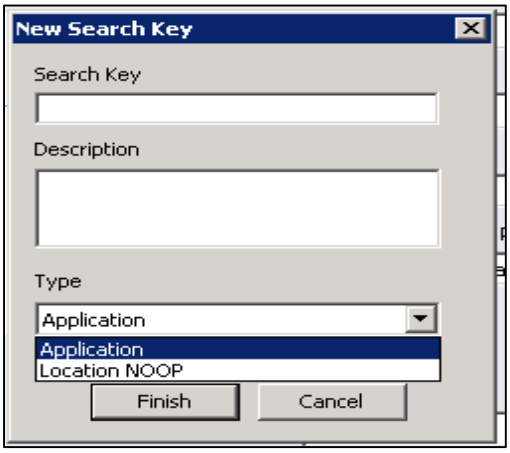
If the search key is created to be used in a number of locations, we can create and store it in the library. However, if the search key is created only for a particular application only, we can create it directly in the application.

2.7.2 Creating Search Key Objects

Following are the steps to create a search key in the Library:

1. Right-click the **Search Keys** heading under **Delivery** and select **New Search Key** from the shortcut menu. The **Create a New Search** Key dialog box opens.

2. Enter a name in the **Search Key** box and, optionally, a **Description**.



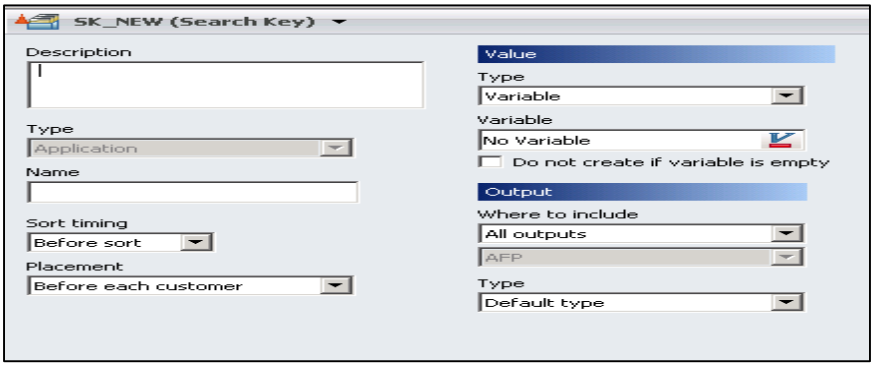
3. In the **Type** drop-down list select one of the following:
   - **Application**- General use with all drivers

- **Location NOOP** – No-operation locator, available with the AFP and Metacode drivers only

4. Click **OK**.
   The search key opens in the Property Panel so as to be defined.

NOTE: *We cannot change the search key **Type** after we have created the search key.*

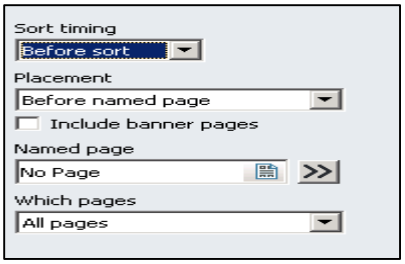
2.7.3 Application Search Keys



- **Type**
  The type drop-down list is inactive since we have already selected the type when we created the search key.
- **Name**
  The name that will appear in the print stream is specified here.
  For AFP output, these are the TLEs that are designed as real searchable keys.
- **Sort Timing**
  The Sort timing drop-down list allows us to select when the search key is to be inserted: **Before sort** or **After sort**. If we select **After sort**, the search key value is populated during post-sort processing.
- **Placement**:
  This drop-down list is used to determine the placement of search key in the print stream. The various options are

1. **Before each customer**
2. **Before each file**
3. **After each customer**
4. **After each file**
5. **Before each document**
6. **Before each bundle**
7. **After each document**
8. **After each bundle**
9. **Before each page**
10. **Before named page**
11. **After each page**
12. **After named page**

- **<u>Before and After Named Page</u>**
  If we select **Before named page** or **After named page** from the Placement drop-down list, additional options become available.



Include Banner Pages

If the named page is a Banner Page, we need to select the **Include Banner Pages** check box to add the search key onto that use of the page object.

Named Page

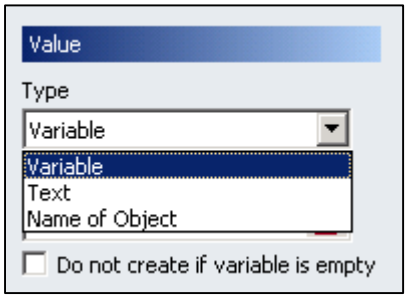
We need to select the Named page to be used from the Select Page dialog box.

Which Pages

Since the same page can be repeated for flow of tables and text boxes, we need to select an option in the Which pages drop-down list for more precise control over search key placement. Options are:

1. All pages—Apply the search key to all instances of the named page.
2. First page—Apply the search key to the first appearance of the named page.
3. Last page—Apply the search key to the last appearance of the named page.

- **Value Area**



Type

In the Type drop-down list the following options are available in order to determine how the value of the key is set.

- **Variable** → A variable is used to set the value of the key. On selecting this option, the **Variable** box is activated which is used to select the variable which will be used as the key.

  Do Not Create if Variable is Empty

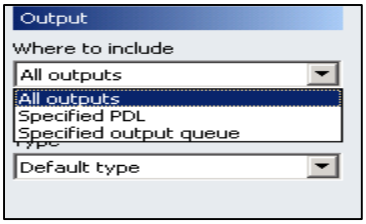
  This checkbox is selected to not create search keys when the value of the variable is null

- **Text** → Any text or combination of characters is used to set the value of the key. On selecting this option the **Text** box is activated where the text to be used is mentioned.

- **Name of Object** →The name of the object where the key is placed (like page name or document name) is **inserted** as the value of the key.

➢ **Output Area**

In the **Output area**, we select on which output devices the search key is placed.



- **Where to include:** In the **Where to include** drop-down list, we select if the search key is placed on all outputs or on specific outputs or queues. Following are its options:
  1. **All outputs**
  2. **Specified PDL**
  3. **Specified Output Queue**

- **Type:** In the **Type** drop-down list, we select how the output driver will interpret the search key. Options are:
  1. **Default Type**
  2. **As Is**
  3. **NOP Comments**
  4. **TLEs**
  5. **NOP Comments and TLEs**

To use search keys in an application, we must add them to the **Search Keys** tab of the application.

2.8 CAMPAIGNS AND MESSAGES

2.8.1 Introduction to Messages

- Messages are text or graphic communications contained in a document or campaign that is placed by the Engine at run time in areas on a page held in reserve by frames.
- All messages can be personalized and targeted with variable data.
- Graphic messages may also include conditional components: text, text boxes, images, charts, and conditional tables with conditional rows and columns.
- We can include a rule that determines whether the Engine includes or excludes the message for each customer. The messages with the highest priority that match the properties for the text frame are selected at Engine run time.

NOTE: *A frame is an object that reserves an area on a page for specific message types and defines what can be placed in that area.*

2.8.2 Types of Messages

- **Text message**: Text messages are one or more words, sentences, or paragraphs of text. They are placed on pages within text frames.
- **Graphic message**: A graphic message can contain formatted text areas and design objects such as images, text boxes, and tables.
- **Insert message**: A pre-printed message stored in an inserter bin that can be added to customer documents during output.
- **Graphic or Insert message**: Messages that are a pre-printed insert or a graphic message, depending on whether the message is defined for the inserter bin.

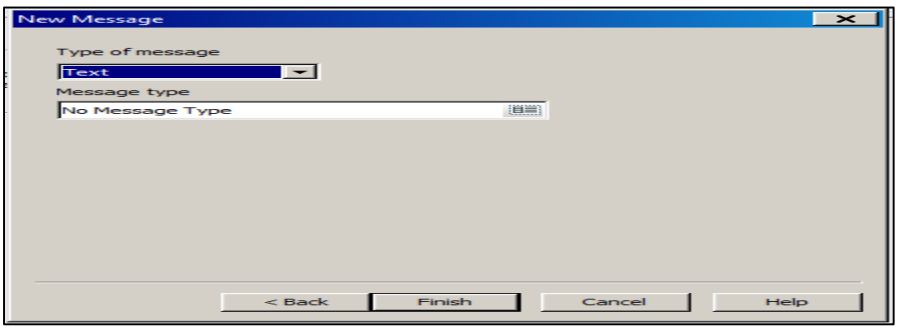
There are two types of text messages:

- **Unformatted** - Created completely in Design Manager and contains a single paragraph style
- **Formatted** - Created in Design Manager or Designer and contains multiple paragraph styles and/or embedded objects.

2.8.3 Creating a Text Message

To create a text message

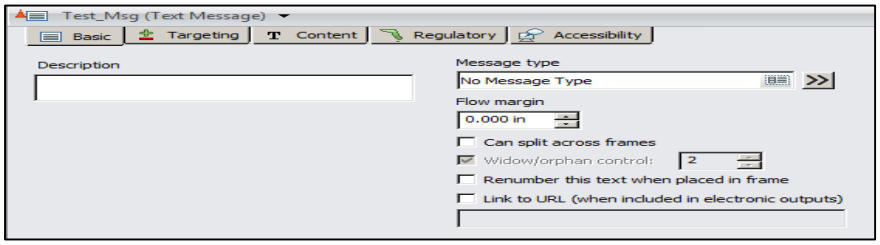
1. Right-click a Messages heading and select New Message
2. New Message dialog box opens
3. Enter a name for the new message in the Name box
4. Enter a description, if necessary, in the Description box
5. Click Next. The New Message dialog box changes.



6. Select Text from the Type of message drop-down list. (Select Graphic to create Graphic message)
7. Select the Text type by clicking the  button.
8. Click Finish.

2.8.4 Message properties

- **Basic Tab:**



- **Description:** Enter a brief description (Optional) about the message.
- **Message type:** It is a category for text messages, change the Text type by clicking the  button. The message type is created under Environment > Design. The message type is used to add styles to the message created.
- **Flow Margin:** The Flow Margin box enables us to set the minimum allowable margin below the current message and before the next when more than one message is placed in a frame.
- **Can Split Across Frames:** When selected, enables messages to begin on one frame and end on another frame if there is not enough room to fit the entire message.

If the check box is cleared, a frame that can fit the entire message must be available or the message is not sent.

- **Renumber This Text When Placed In Frame:**

Select the Renumber this text when placed in frame check box to force the text to adopt the numbering pattern of preceding text.

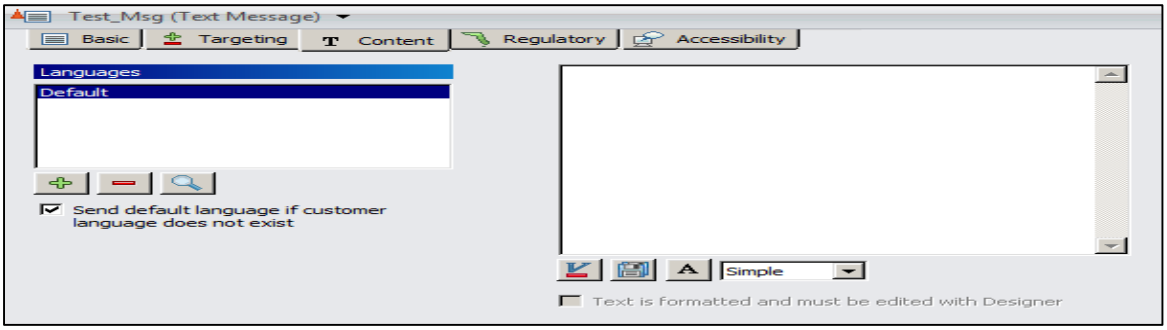
If we have multiple numbered messages and clear this check box, each message is preceded with the number one.

- **Link to URL (When Included in Electronic Outputs):**

The Link to URL (when included in electronic outputs) is used to link a message to an Internet site or an email address when used with an HTML or other eDriver (like PDF). When we select this check box, enter an email address or URL in the box below.

NOTE: If we are working in Designer, save and close the message before we edit the message properties in Manager, or we may overwrite changes we wanted to save in Designer.

- **Targeting Tab:** The Targeting tab enables us to place selection rules on messages and to track the customers who receive the message.
- **Content Tab:** The Content tab enables us to view and edit the message text.



- **Languages Area:** The Languages area is used to distribute the message in the customer's native language. This uses the Languages defined under environment properties.
- **Text Display Area:** enables us to view the content of a paragraph. If we are creating an unformatted paragraph, we can type or import the text file (*.rtf or *.txt format) into this area.
- Variables can be inserted at the cursor position using the button. It opens the Variable palette where we can select the variable we want to insert.
- Text to insert text from existing RTF for TXT files, click the button.
- Font Click the button to select a single font to use for the entire paragraph. The Select Font dialog box opens where we can select the font and its size.

We can also change the type of paragraph we are creating. Options are:

- **Simple** - Lines of text (no bullet or number). This is the default setting.
- **Bullet** - Bulleted text.
- **Number** - Sequentially numbered text.

NOTE: *The buttons under the unformatted text display are not available for formatted text.*

- **Text is formatted and must be edited with Designer** check box is common to unformatted and formatted paragraph. It is always inactive since it is automatically selected by Exstream. If it is selected, it means all formatting must be done in Designer. If it is cleared, text editing can be done in Design Manager.

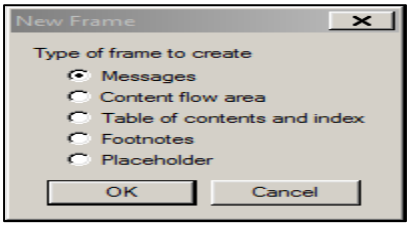
- **Regulatory Tab:** The Regulatory tab is used to enable jurisdictions and effectively, which control distribution to customers based on their location and the date when output is created. Compliance support module is needed to use this option.

2.8.5 Message frames

Messages are dynamically placed in frames. This means the placement of messages can change according to the frames they are placed in and which messages are chosen by priority and targeting rules.

2.8.5.1 Creating Message frame

1. Click the button.
   The New Frame dialog box opens.

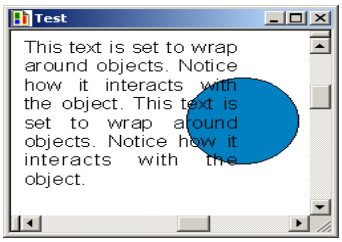


2. Select the Messages radio button.
3. Click OK. The Insert frame dialog box opens.
4. Select the required options
5. Click OK.

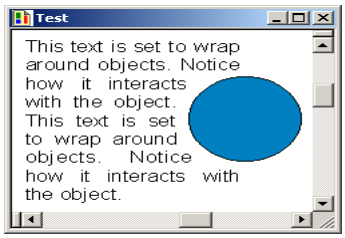
The frame is placed on the page.

2.8.6 Message Frame Properties

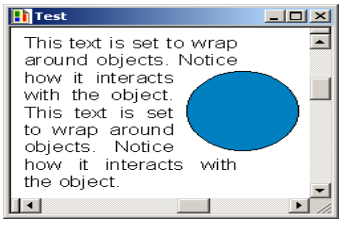
- **Message Frame:** The options on this tab enable us to select which messages are accepted
  - **Fill Order:** The Fill order box is used to change the order in which frames are filled. The fill order is initially determined by the order in which the frames are created, but we can use this box to change the order. If we change the order, the fill order of other frames is updated, as well.
  - **Border:** We can place lines around the frame. Click the side where we want to place a line.
  - **Wrap Around Overlapped**
    The Wrap around overlapped drop-down list determines how text in the frame flows around static objects on the page. Options are:
    - None - Text does not wrap
    - Fit - Text follows the shape of the other object
    - Rectangular - Text maintains a rectangular shape when wrapping around objects



(NONE)



(FIT)



(RECTANGULAR)

  - **Allow Multiple Messages in Frame:** Select this check box if we want if more than one message is to be placed in the frame. If the check box is cleared, only one message is placed in the frame, even if there is space for other messages in the frame.
  - **Message Flow Across Frames**: The Message flow across frames drop-down list controls the flow of messages from frame to frame and defines if objects can flow in or out of the frame. Options are:
    - **None** - No flow in or out of the frame. The contents must fit in the frame or it does not appear/included.
    - **In** or **Out** - Contents flow into this frame, and out into another.

- **In Only** - Contents flow into this frame only.
- **Out Only** - Contents flow out of this frame only.

❖ **Text Messages:** Select the Text messages check box to allow text messages in the frame. We also need to select a type of message from the Allowed message type drop-down list. Select Any to allow all message types in the frame, or a specific type of message.

❖ **Graphic Messages:** Select the Graphic messages check box to allow graphic messages in the frame. We also need to select a Primary template.
The template we select determines the size of our frame. We can make the frame larger, but if we make the frame smaller, our graphic message may not be placed.

NOTE: *The Message template can be created under Environment > Design > Templates > Message templates in a similar way creating paper types.*

❖ **Alternative Contents:** Use the Alternative contents drop-down list to select how other messages are placed in the frame. Options are:
- **None, must match template** - Only messages with the specified template are placed in the frame.
- **Same message types** - Messages with different templates can be placed in the frame if they have the same message type.
- **Anything** - Any message with any message template can be placed in the frame.

❖ **Allow Smaller Messages:** Select the Allow smaller messages check box to allow messages whose dimensions are smaller than the primary template in the frame. If the check box is cleared, only messages that are the same size or larger can be placed in the frame.

❖ **Placement Tab:** In the Placement tab we select various placement and position properties for frames.

❖ **Dynamic Size and Placement Tab:** Dynamic Whitespace frames allow frames to be composed at the end of processing and be changed or personalized to each customer.

2.8.7 Introduction to Campaigns

Campaigns are Exstream object which are used to group together multiple messages with similar purpose. Campaigns can include messages printed on the customer document or pre-printed messages that are added with an inserter. The same campaign can be used in multiple applications.

Special features include the ability to:

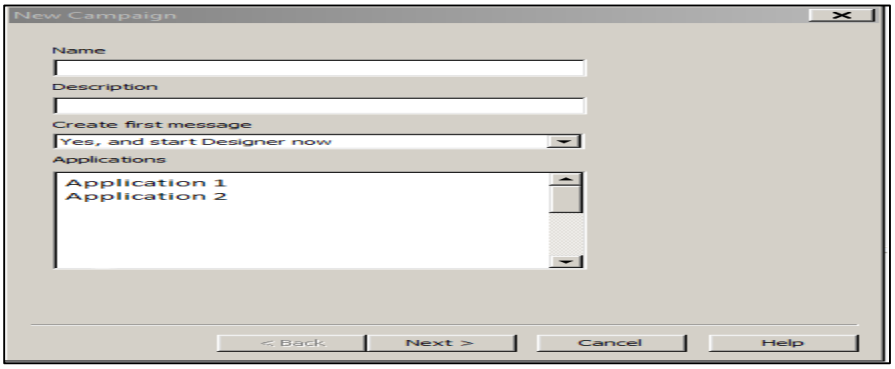
➢ Control the total number of times and the frequency a campaign is sent.
➢ Simplify the implementation of customer information delivery systems.
➢ Prioritize for each customer (send the most important message that fits in the space available).
➢ Target messages based on qualification rules and parameters.
➢ Design campaigns in a media-independent environment. For example, we can design a message in Photoshop and include it in our Exstream applications.

- Develop one-to-one marketing campaigns – unique content that is personal, intimate, and focused.

2.8.8 Creating a new Campaign

To create a new campaign

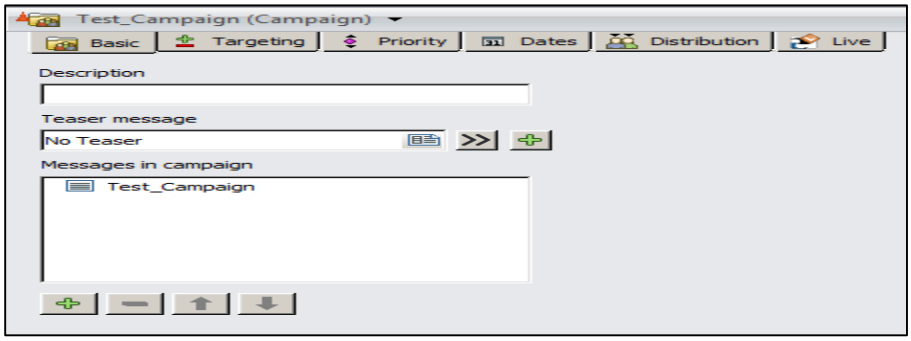
1. Right-click the Campaign heading in any folder and select New Campaign from the shortcut menu. The New Campaign dialog box opens.



2. Enter the name of the campaign in the Name text box and a description in the Description text box (optional).
3. From the Create first message drop-down list, select one of the following options:
   - **Yes, but do not design the message now** - Message is created with the same name as the campaign, but it is not opened in Designer.
   - **Yes, and start Designer now** - Message is created with the same name as the campaign. It is opened in Designer.
   - **No, will add messages later** - Only the campaign is created.
4. Select an application from the Applications list.
5. We can choose more than one application. The campaign is automatically included in the application(s) we select from the list.
6. Click Next.

NOTE: *If we select either Yes option from the Create first message drop-down list, the campaign creates only one message for a campaign. To add more messages, add them to the campaign after it has been created.*

2.8.9 Campaign Properties

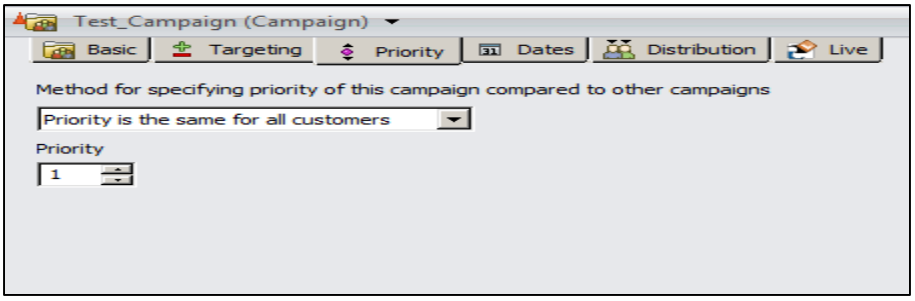


- **Basic Tab:** The Basic tab enables us to determine the messages available in the campaign.

- **Description:** Enter a brief description (Optional) about the campaign.
- **Teaser Message**: The teaser message can be any Exstream message. There can be only **one** teaser message for a campaign. A teaser message is a message that normally prints on the first page of a document that is designed to direct customers to a location within the document.
- **Messages in Campaign**: The Messages in campaign box lists the messages that are in the campaign. We can add and remove messages by clicking the  or  buttons.

- **Targeting Tab:** The Targeting tab enables us to target customers based on a combination of rule requirements, an external data list, and previous campaign activity.
  - **When Customer Receives Multiple Documents:** This drop-down list enables us to control the placement of campaigns within a multi-document application. Options are:
    1. **Normal, use once for entire customer** – This is the default option. If a campaign message qualifies, it is placed in the first allowed position, no matter how many documents are being sent or how many times the customer qualifies.
    2. **Repeat campaigns on each document** – This enables us to place the same message on each document in an application. This means if a campaign is qualified for a customer, and one message is sent, then that single message appears in every document.
    3. **Use as qualified on section documents** – This option is only used in section-driven documents. If using this option, the timing for qualification must be set for Each Section, Until Qualified or Each Section, Sum Qualified.
  - **Campaign-Driven Pages:** The Campaign-driven pages drop-down list controls whether additional pages can be created in a document for the campaign contents.
    Options are:
    1. **Do not create page for this campaign** - Send this campaign only if there is room on available pages.
    2. **If no additional postage** - Includes the campaign if adding this campaign adds another page to the envelope but does not increase postage.
    3. **Up to maximum marketing pages** - Sends the campaign even if adding this campaign adds another page to the envelope and increases postage.
    4. **Always (campaign must be sent)** - Send this campaign even if it adds another page to the envelope, adds additional postage, or creates more marketing pages for the customer than the specified maximum.
    5. **Content only** - Send this campaign only if it is placed in a content frame. Do not use this setting if the campaign's messages can appear in both Whitespace and Content frames.
  - **Send to Customers Listed in Reference File:** Once we select the check box, choose the file from the drop-down list. Usually reference files are used for list-based targeting. If the key appears in the file, as defined in the data file properties, the record is included.

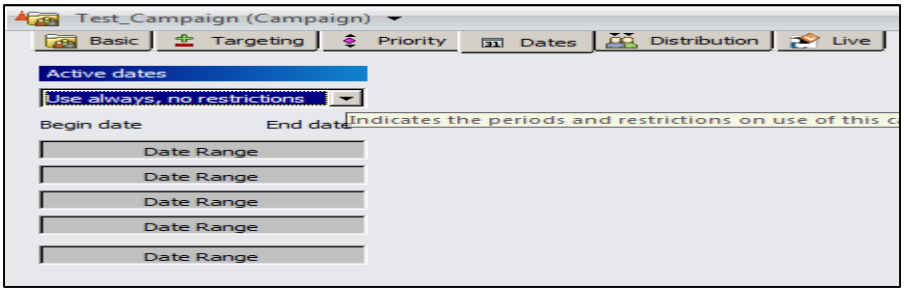
➢ **Priority Tab:**



Choose one of the following from the Method for specifying priority of this campaign compared to other campaigns drop-down list:

- **Priority is the same for all customers** - Sets the priority of the campaign to a specific number. Use the Priority box to select the number representing the priority of the campaign.
- **Priority is set by a rule for each customer** - Bases the priority of the campaign on a rule. An area appears where we can enter the rule.

➢ **Dates Tab:**



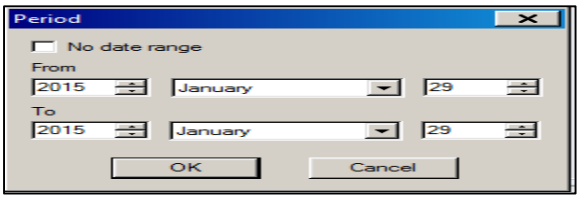
We can choose specific dates during which to send or not send the campaign. For example, a coupon campaign may expire after three months.

❖ **Active Dates**

Select one of the following from the Active dates drop-down list:

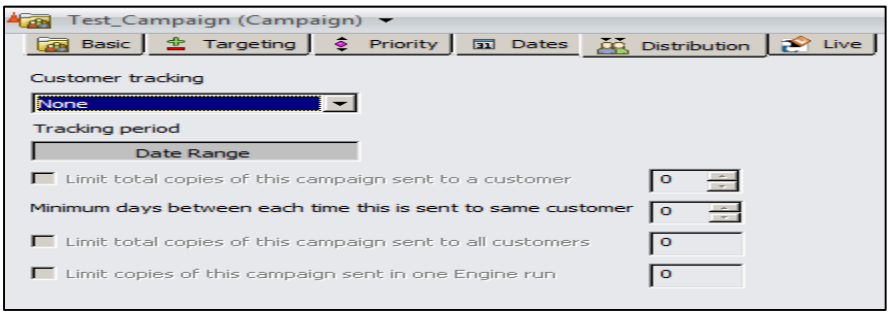
- **Use always, no restrictions -** Without any time restrictions.
- **Use during periods below -** Only during the times we specify.
- **Use except during periods below -** Always except the times we specify.

When selecting Use during periods below or Use except during periods below from the Active dates drop-down list, click on the first usage period box to display the Period dialog box, as shown in the following figure. Complete the dates.



To remove a period, select the **No date range** check box.

➢ **Distribution Tab:**



❖ **Customer Tracking**

The Customer tracking drop-down list enables us to record information about the paragraphs sent and, if selected, the customers that receive the paragraphs. Options are:

- **None -** No tracking is performed.
- **Summary -** Information about paragraphs, but not each customer is recorded.
- **By Customer -** Information about customers and paragraphs is recorded.

❖ **Tracking Period:** If we select Summary or By Customer, the Tracking period box becomes available. This enables us to limit the tracking to a certain time period. Click the box to open the Period dialog box.

❖ **Limit Total Copies of this Campaign Sent to a Customer:** Select the Limit total copies of this campaign sent to a customer check box and enter a number in the box to the right to specify the maximum number of time the campaign can be sent to a single customer.

❖ **Minimum Days between Each Time This is Sent to Same Customer**: If we select any number higher than one, the Minimum days between each time this is sent to same customer box becomes active. Enter the number of days we want between each instance of the campaign the customer receives.

❖ **Limit Total Copies of This Campaign Sent to All Customers:** Select the Limit total copies of this campaign sent to all customers check box and enter a number to limit the total number of times a campaign can ever be sent.

❖ **Limit Copies of This Campaign Sent in One Engine Run:** Select the Limit copies of this campaign sent in one Engine run and enter a number to limit the number of times a campaign can be sent at a single time.

Do Not Send to Customers after Response Select the Do not sent to customers after response check box to ensure the campaign is not sent to customers who have responded.

2.9 BARCODES

A Barcode is a way to represent data so that it can be read by barcode tracking equipment. Barcode can be printed on the page in two ways:

1. Directly placing the barcode on the page as barcode component.
2. Placing the barcode in the inserter, adding the inserter in an output queue, the documents that are printed using this output queue will be printed with the barcode added.

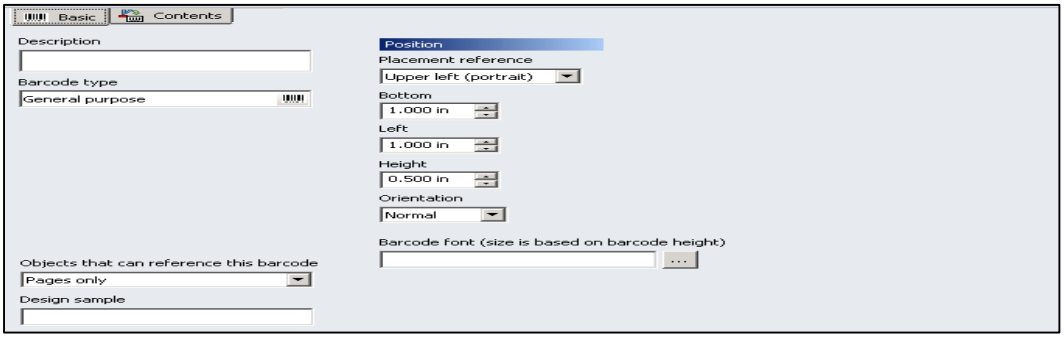
2.9.1 Ways to Create Barcodes

There are two ways to create barcodes:

2.9.1.1 Creating a Barcode in Design Manager

1. Under **Library,** right-click the **Barcodes** heading and choose **New Barcode** from the shortcut menu.
2. Type a **Name** for the new barcode and give a short **Description** (Optional).
3. Click **Next**.
4. Select the type of barcode from New Barcode list. Click Finish.

The barcode object opens in the Property Panel to define.



2.9.1.2 Defining a Barcode

➢ **Basic Tab:**

❖ **Barcode Type:** In the Type drop-down list, select the barcode format for this object. The properties and options will be changed according to the barcode type. Following are the different types of barcodes available.

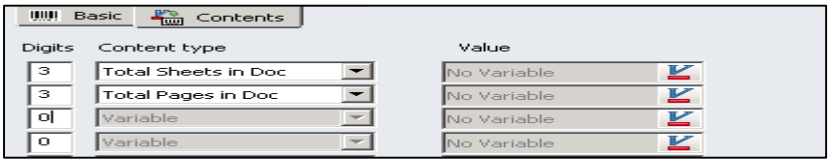
1. OMR	2. General Purpose	3. POSTNET
4. GBR OMR	5. Japanese Postal	6. EAN 128
7. 3 of 9	8. Interleaved 2of5	9. PDF417
10. Data Matrix 2D	11. Code 128	12. Modified Plessey
13. UPC	14. QR Code	15. Four State Barcode
16. EAN- 8		

❖ **Position Area**

- Specify the distance from bottom, left in the corresponding **Bottom** and **Left** box.
- In the **Height** box, specify the length of the longest bars for this code.
- Barcode will be printed on the page with the specified orientation and position.

- **Objects that can reference this barcode**
  We can limit the objects that can incorporate this barcode. Choose from:
  - Messages Only
  - Pages Only
  - Templates Only
  - Messages, Pages, Templates
  - Inserter Only
- **Barcode Font:** Select the font we want to use when creating a barcode.

➢ **Barcode Contents Tab**
All barcode types support a Contents tab.



Some of the commonly available Content Type options are mentioned below.

1. Empty	2. Total Sheets in Doc	3. Sheet in Document
4. Page in Stream	5. First Sheet	6. Not First Sheet
7. Not Last Sheet	8. Sheet in Queue	9. Document in Break
10. Variable	11. Total Pages in Doc	12. Page in Document

- Variable- select a particular variable in the final column by clicking the button and selecting the one we need from the Variable palette.
- Sheet, Page, and Document options enable us to base barcode content on its position at run-time.
- Bin, followed by the bin number, enables us to base barcode content on a particular bin at run-time.
- Choose Empty for a bar that has no value (not used).

2.9.1.3 Create Barcodes in Designer

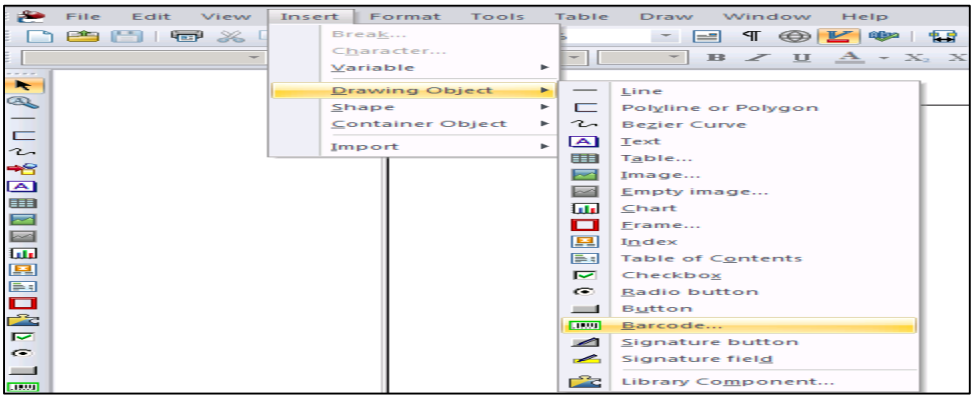
Create Barcodes from Fonts
We have the option of using Exstream Designer to create barcodes with a barcode font. This method sets barcode information to the value of a variable.

For Example-
In an address block we insert the variables for customer name, address, city, State and zip code. On the next line we can insert the zip code variable again and format it with the Barcode font for example (USPS barcode font). When we run the Engine, Exstream creates a POSTNET barcode that changes with the zip code for each customer.

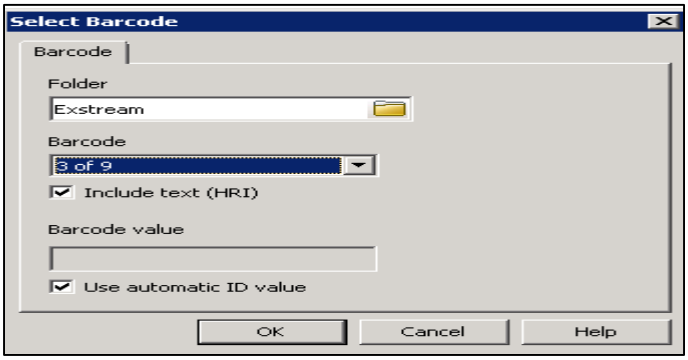
Robert Stevens
1200 Main Street
Lexington, KY 40507

2.9.1.4 Steps to add a Library Barcode to an Object in Designer

- Click on the **Insert** menu in Designer and choose Drawing Object.
- Choose **Barcode** from the submenu.

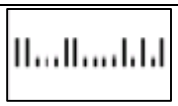
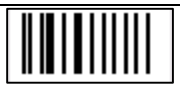
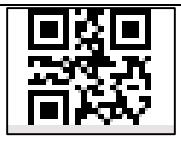


In the **Barcode** drop-down list choose the barcode we need:



1. Select the **folder**.
2. In the **Include Text (HRI)** check box, select the check box to enable Human-Readable Identifier text, this option is enabled so that human readable texts are printed along with the barcode.
3. Under Barcode value:
   - If we want Exstream to provide a Tracking ID for this code, select the **Use automatic ID value** check box.
   - If we want to specify a Tracking ID for this barcode, clear the check box and type the ID in the above text box.
4. Click **OK**.

2.9.2 Different types of barcodes

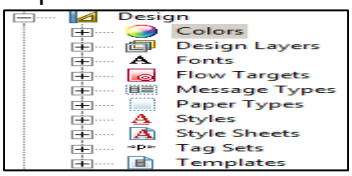
NAME	EXAMPLE	DESCRIPTION
1. OMR		A barcode that is often used to control inserters, collect data, and for inventory control.
2. General Purpose		A versatile barcode that can be defined according to the specifications of a particular barcode reader.
3. POSTNET		Conforms to the standards set by the United States Postal Service and can contain ZIP Code or Delivery Point Codes. Also called USPS barcode or 3 of 5 barcode.
4. GBR OMR		A type of OMR barcode that uses a combination of broken and solid lines.
5. Japanese Postal		A 21-character, font-based barcode that complies with the requirements set by the Japanese Postal System.
6. EAN 128		A shipping barcode similar to a Code 128 barcode that enables scanners and processing software to automatically discriminate between EAN-128 and other barcode symbologies.
7. 3 of 9		A widely used barcode that can be decoded with virtually any barcode reader. Also called 39 or Code 39 barcodes.
8. Interleaved 2of5		Often used in warehouse and industrial applications, this barcode consists of five bars to denote two numeric characters.
9. PDF417		A two-dimensional, stacked barcode that can encode large amounts of data, such as the content of a shipping manifest or equipment maintenance history.
10. Data Matrix 2D		Two-dimensional barcodes that have greater capacity for storage than standard barcodes, which make them especially useful for marking small items. Also called 2D DataMatrix barcodes.
11. Code 128		A versatile barcode used in many different industries, which supports the encoding of all 128 ASCII characters.
12. Modified Plessey		Primarily used for inventory purposes, the length of this barcode can vary between applications.
13. UPC		A 12-digit barcode used to identify unique products for retail or tracking purposes.
14. QR Code		Two-dimensional barcodes that have greater capacity for storage than standard barcodes. Often used when a document might be scanned or copied, or to embed Web addresses to be read by mobile phones.
15. Four State Barcode		Used for sorting and tracking mail and complies with the regulations set by the United States Postal Service. Also called the Four-State Customer Barcode (4CB).
16. EAN- 8		A barcode similar to a UPC barcode, yet specifically designed to use as little space as possible. Therefore, it is often used on small packages.

2.10 ENVIRONMENT

2.10.1 DESIGN ENVIRONMENT

There are many features which we can set up in the environment tab, below we will discuss about the “design” tab.

Inside the Design tab there are many functionalities which we can setup according to the requirement. Below is the screenshot of all the functionalities under Design tab:

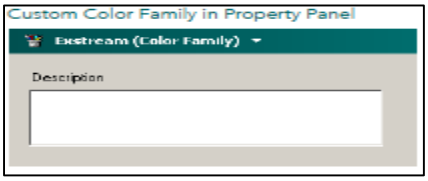


2.10.1.1 Colors

This feature is useful if our company standardizes certain colors for logos, fonts, and other publication elements. We can create any number of different families with specific colors and their names as defined by our Marketing department.

➢ **How to create Color Families :**

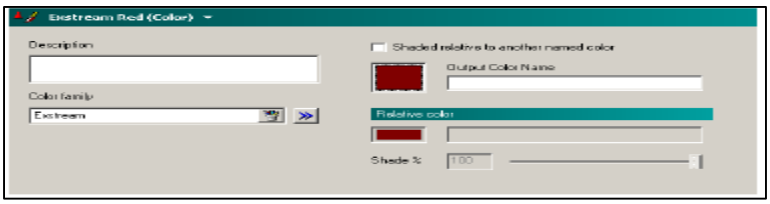
- Open the **Environment** heading, then the **Design** subheading as shown.
- Right click on **Color Families**. The Color Family shortcut menu appears.
- Choose **New Color** Family. The Color Family dialog box appears.
- Type the **name** of the new color family and click **Finish**.



➢ **How to insert a new custom color into the color family :**

- Right click on the new custom color family which we have created and select the option **Insert new color** as shown below :

- Name the **new color** and click **Finish**.
- Open the new color in the property panel and adjust the color well by sliding or typing values for the color as shown below :



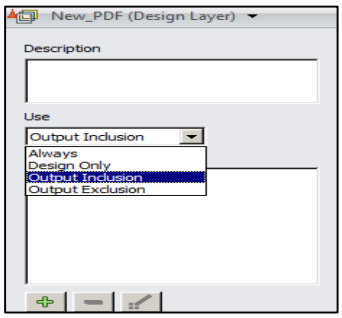
2.10.1.2 Design Layers

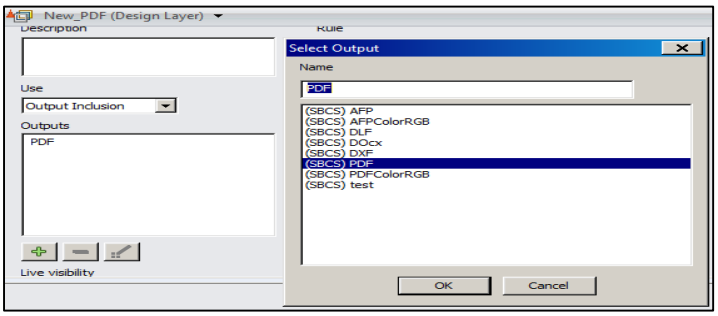
Design Layers are associated with specific output devices (For example PDF, AFP, Metacode outputs and so on). Depending on how the design layer is configured, design objects are included or excluded only to the specific output devices which it is configured to.

➢ How to create a design layer :

- Right click on the **design layer** tab and click **New Design Layer** and enter a **name** and press **OK**.

- In the property panel, add the output to be included or excluded as shown :

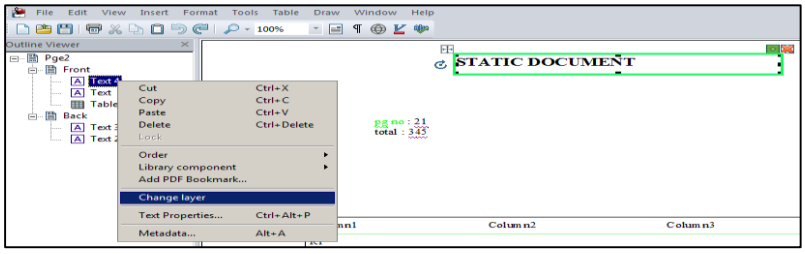




- Click **OK**. Now the design layer for PDF inclusion is created.
- In a similar manner, we can create design layers for different outputs and configure to include or exclude.

➢ How to use the design layer on a design object :

- Right click on the design object and select **change layer** and select the **design layer** we want to implement and press **OK**.

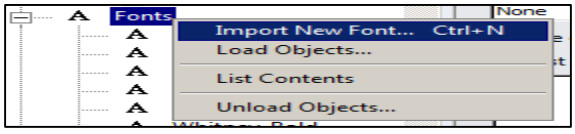


- So the above design object is written into the design layer for which the output is affiliated to.

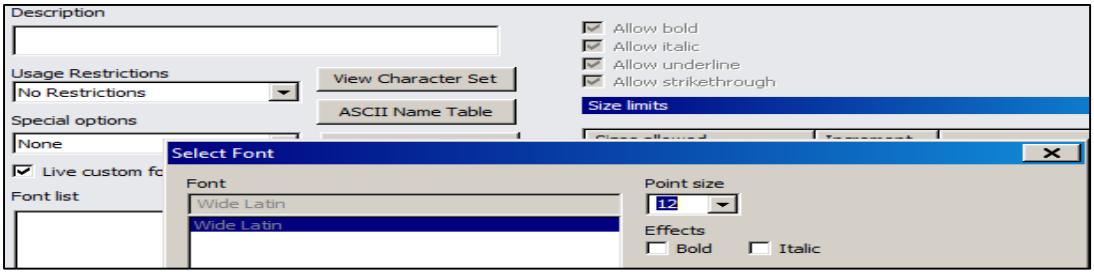
2.10.1.3 Fonts

Font tab is used to create a new font or customize the font.

- Right click on the **font** tab and select the option **import new font** and select the type of font from the font list in the property panel.



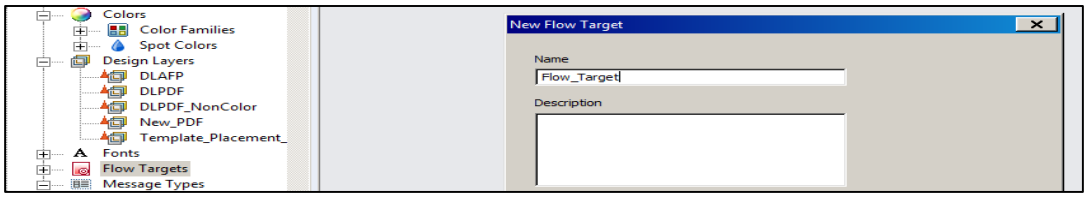
- In the property panel of the font add the font size to be used.



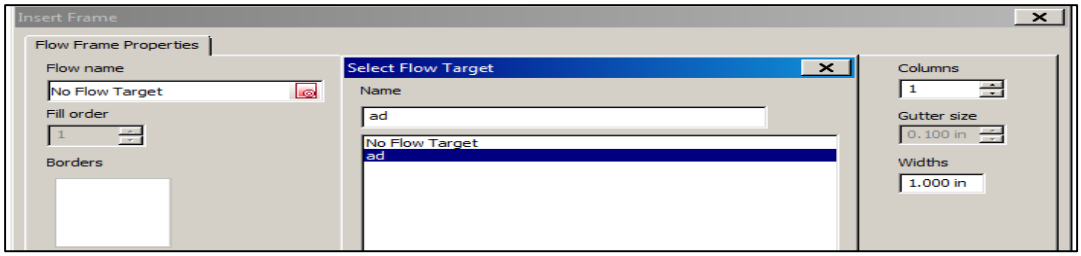
2.10.1.4 Flow Targets

Whenever there are flowing objects on a page and the flow page holds multiple flow frames and we want to target the flow to a particular frame, then flow target comes into picture.

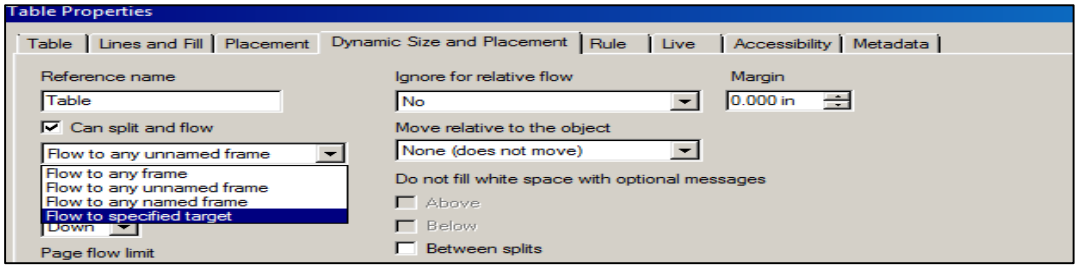
- Right click on the **flow target** tab and create a **new flow target** and press **OK**.



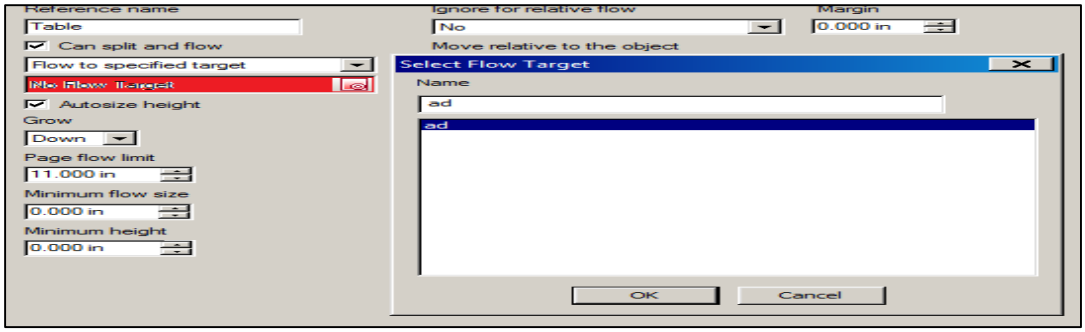
- In flow frame properties specify the **flow target** in the **flow name**.



- Mention the **flow target name** in the object which is flowing as shown in the screenshot below. Click on the **Dynamic Size and Placement** tab and check the option "**Can split and flow**" and the select the "**Flow to specified target**" option from the drop down.



- Choose the **flow target list** and press **OK**. Now this particular design object will flow only to the Flow target selected from the list.



2.10.1.5 Message Types

The basic requirement for creating messages is to create a message type. Message type defines the basic layout of a message. Message types are only used for text messages, whereas Graphic messages use a specific template.

For more information on message types, please refer to campaigns and messages section.

2.10.1.6 Paper Types

A paper type defines the property of a paper to ensure the pages are printing on the correct page. When we create a page we must specify the appropriate paper type.

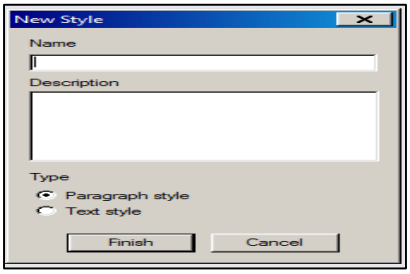
For more information on paper types, please refer to 2.1.4 section.

2.10.1.7 Styles and Style Sheets

➢ Styles:

To create a style the same steps are followed as those followed to create any other object in Exstream.

- ❖ Right-click the **Styles** heading.
- ❖ Choose **New Style** from the shortcut menu. The Create a New Style dialog box opens.
- ❖ Enter a **name** for the style in the Name field.
- ❖ If we want to add a description of the style, type it in the **Description** field.
- ❖ Select the radio button for the Type of style we are creating.
- ❖ Click **OK**. The new style opens in the Property Panel for us to define.



❖ Type :

There are two options for the type of style we are creating. Choose the radio button next to the type we want to create.

1. A **Paragraph style** defines formatting and spacing properties for an entire paragraph.

2. A **Text style** defines the formatting of a selection of text in a paragraph, even if that entire paragraph has been defined with another Paragraph Style. We can use Text Styles in cases where we want to control the formatting for part of a paragraph.

➢ Style Sheets:

A style sheet is a collection of styles as they are used for a publication. It is a collection of Paragraph styles and Text styles that can be applied to a page. Once a style sheet is applied, all the styles assigned to that style sheet are available for use on the page.

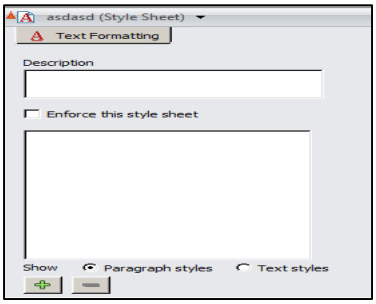
In Exstream, we define the name for the style, but the actual properties are defined for each style in the style sheet. Each style is a unique entity in the style sheet for each publication.

❖ Creating style sheets :

- Right-click the **Style Sheets** heading.
- Choose **New Style Sheet**. The New Style Sheet dialog box opens.
- Type a **name** and also the **description** for the style sheet in the respective fields.
- Click the **Finish** button. The style sheet opens in the Property Panel for us to define.

❖ Style sheet properties:

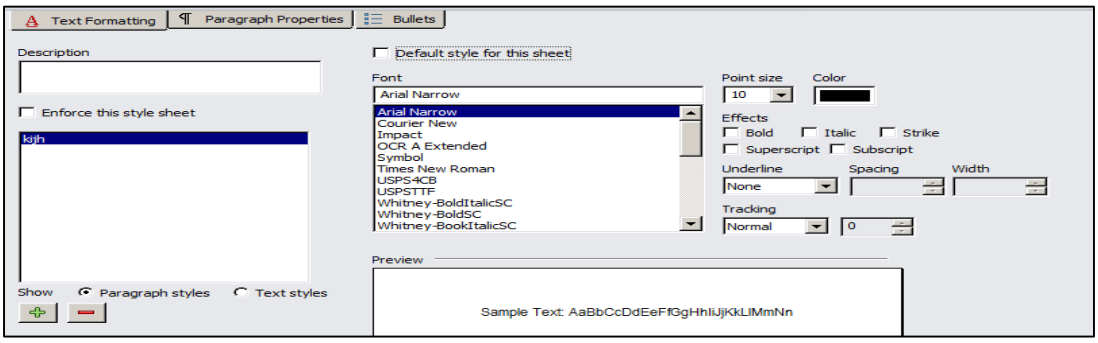
When a new Style Sheet opens in the Property Panel, the only tab available is the **Text Formatting** tab. As we add styles to the style sheet additional tabs and options appear.



- **Enforce this Style Sheet:** The **Enforce this style sheet** check box, when selected, forces the user to adhere to the formatting of the styles available in this style sheet.
- **Show:** The Show radio buttons control the style type we are adding to the style sheet. If the Paragraph styles radio button is selected, only paragraph styles appear in the Select Style dialog box. In addition, when styles are assigned to the style sheet, these radio buttons control the styles visible.
- **Adding a style:**
  1. Select the radio button next to the type of style we want to add.
  2. Click the  button to add a new style. The Select Style dialog box opens.
  3. Select the style we want to add from the Select Style dialog box.
  4. Click **OK**.

The text formatting options appear on the right side of the tab. If we add a paragraph style to the style sheet, the Paragraph Properties tab appears.
Follow the above steps to add as many styles as necessary to the style sheet.

❖ Text Formatting tab

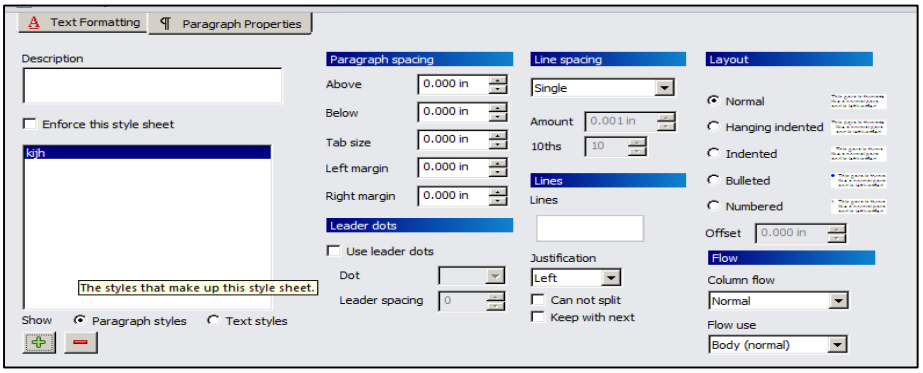


Font Formatting :

- We can also control how the font looks by adding bold, italic, underline, strike-through, along with others.
- The Underline drop-down list is where we specify how the underlining appears on the page. It can be None, All or Words only.
- We cannot control the underline of URLs, these appear with the entire selection underlined, including spaces. This is the only way URLs appear.
- We can also control the tracking, or character spacing, of the font. From the Tracking dropdown list, the options are None, Condensed or Expanded.
- **Default Style for this Sheet:** The Default style for this sheet check box, when selected, uses the current style as the default text formatting when a new text box is added to the page with this style sheet applied. This check box is only active when we are specifying a paragraph style.

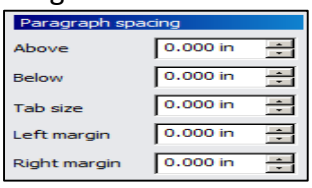
❖ Paragraph Properties Tab

The Paragraph Properties tab is only available if we are editing a paragraph style. This tab offers formatting options for the paragraph in the style.



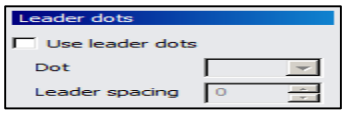
- ### Spacing

  The Spacing area is where we can define the space above and below a line, as well as tabs and margin sizes.



- **Leader Dots**

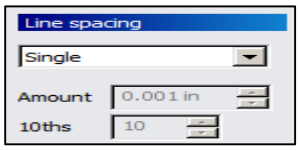
In the **Leader Dots** area, we can specify leader dots to be used. These dots appear when we press the TAB key.



1. **Use leader dots:** Select the **Use leader dots** check box if we want to have leader dots available with this style. This check box is cleared by default.
2. **Dot:** When we select the Use leader dots check box, the **Dot** and Spacing boxes become available. In the Dot box, we specify the character we want to use as the leader dot. We can also type a custom leader dot into the box. A space is the default character.
3. **Spacing:** The **Spacing box** is where we specify the amount of space between each character in the line of leader dots. Each number represents a tenth of a point.

- **Spacing Area**

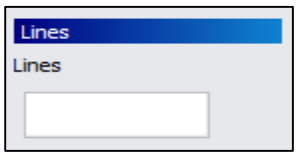
The Spacing area controls the amount of spacing between lines in the paragraph.



We can choose **single** line spacing, **space and half**, **Double**, **Minimum**, **Exactly** or **10th of space**.

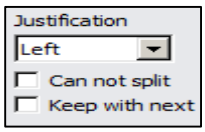
- **Lines**

The **Lines** area allows us to add a ruled line above or below the paragraph. To add a line above, click the top of the box. To add a line below, click the bottom of the box. To remove the line, click the line.



- **Justification**

The Justification area is where we specify how the text is justified in the text box.



Choose the justification for the text from the Justification drop-down list. The options are: **Left** (default setting), **Right, Center or Justify**

1. **Can Not Split:** The **can not split** check box, when selected, does not allow the text in the paragraph to split across columns, frames, or pages. This check box is cleared by default.
2. **Keep with Next:** The **Keep with next** check box, when selected, keeps the current paragraph with the following paragraph.

- **Column Flow**

  The Column flow drop-down list specifies how the text is handled over columns. The choices on the drop-down list are:

  1. **Normal** – The text is handled normally, filling one column before filling the following column. This is the **default** setting.
  2. **Spans columns** – the text spans across all columns, creating a single line of text across multiple columns. A headline is an example of text that spans columns.

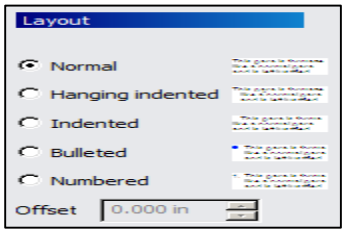
- **Flow Use**

  The Flow use drop-down list specifies the use of the paragraph as a header or footer. The options in the drop-down list are:

  1. **Body** – the text is normal body text. This is the default setting.
  2. **Header** – the text is a header row, which traditionally appears above the body text.
  3. **Repeating header** – the text is a header row which repeats on each page the text flows to.
  4. **Repeating except first** – the text is a header row which repeats on each page the text flows to, except the first.
  5. **Footer** – the text is a footer row, which traditionally appears below the body text.
  6. **Repeating footer** – the text is a footer row that repeats on each page the text flows to.
  7. **Footer except last** – the text is a footer row that repeats on each page the text flows to except the last.

- **Layout**

  The Layout area controls the way the text appears on the page.



The options are:

1. **Normal** – the text appears as normal, with no indents. All the text is flush left (unless otherwise specified by Justification).
2. **Hanging Indented** – the first line of text appears as normal and all subsequent lines of text in the paragraph appear indented from the first. We must specify the Offset measurement.
3. **Indented** – the first line of text is indented from the rest of the paragraph. We must specify the Offset measurement.
4. **Bulleted** – the paragraph appears with a bullet preceding it. We can specify the bullet type and color on the Bullets tab that appears when we choose this option. Use the Offset to specify the amount of space between the bullet and the text.
5. **Numbered** – the paragraph appears with a number preceding it. We can specify the numbering style, font, and color on the Numbering tab that appears when we choose this option. Use the Offset to specify the amount of space between the number and the text.

- **Offset**

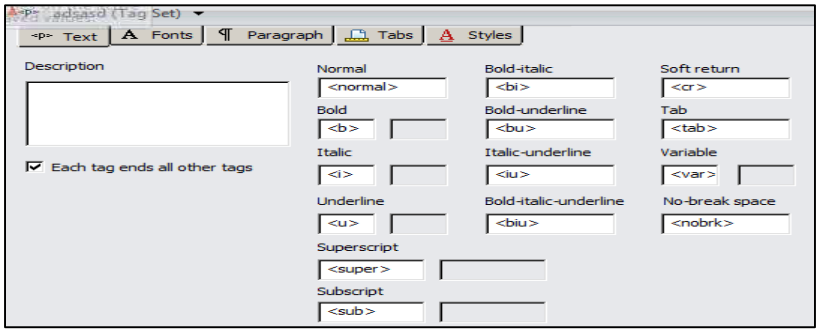
  The measurement we specify in the Offset box controls the size of the out dent, indent, or space between numbers/bullets and text. This box is active if we choose any option other than Normal from the Layout options.

2.10.1.8 Tag Sets

Dynamic File Import Module also enables us to define customized tag sets for use with Dynamic File Import features. We can make changes to the default tags.

➢ **Create New Tag Set Object :**

- Right-click on the **Tag Sets** heading in the Library.
- Select **New Tag** Set.
- Type a name in the **Name** dialog box. Type a **Description**, if desired.
- Click **Finish** to create the tag set.



The new object opens in the Property Panel for us to define.

The Tag Set properties have five tabs: Text, Fonts, Paragraph, Styles, and Tabs.

➢ **Tag Set Properties - Fonts Tab**

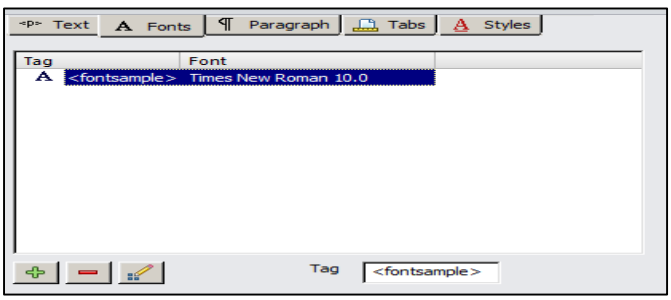
We can change fonts in the flow of our text with font tags. HP Exstream automatically packages the fonts referenced by these tags.

❖ **Add Font Tag**

To reduce keystrokes, we make changes to Exstream default tags.

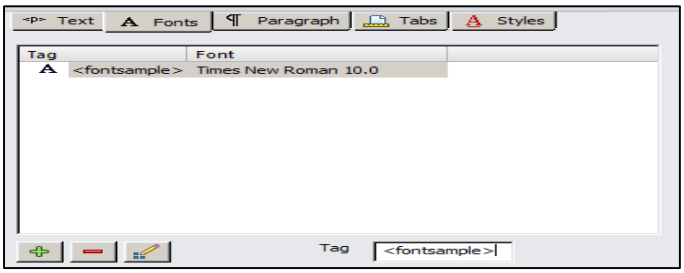
- Click the Fonts tab.
- Click the  button.

HP Exstream creates a Tag in the text box area. The default naming convention is <font1> followed by <font2>. Exstream displays our default font and font size under the Font column. The system administrator set these defaults in the Design Defaults tab of System Settings.



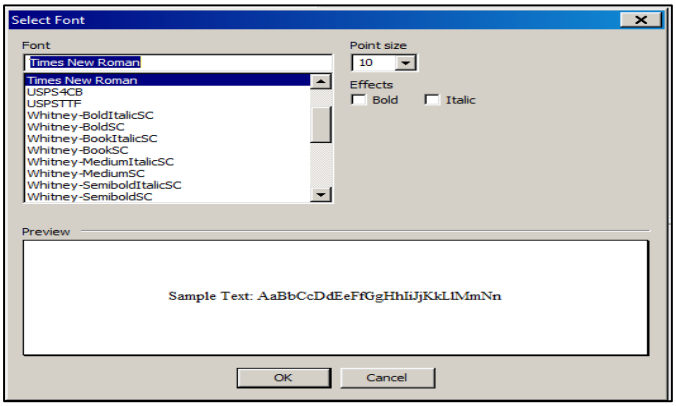
❖ Edit Font Tag

Highlight the font tag we want to change and Type the new name to the font into the Tag text box.



❖ Edit font information :

Highlight the font tag we want to change and double click on the font to change the properties of the font.



Make the changes to this dialog box as needed.

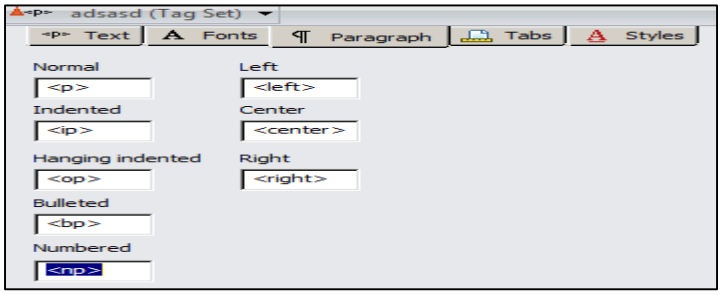
- To change a font, select a new font from the top drop-down list.
- To change the point size, select a new size from the Point size drop-down list.
- We can set the font to always appear bold, italic, or both with the Bold and Italic check boxes.

❖ Delete Font Tag :

Highlight the font we want to delete and press the button to delete the font type.

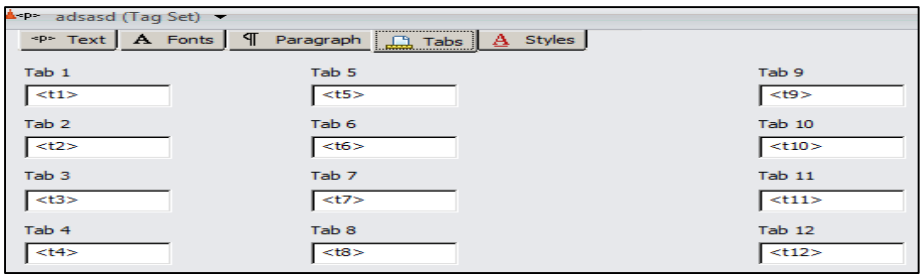
➢ Tag Set Properties - Paragraph Tab

The tags in this tab affect the formatting of paragraphs. The first column of options affects the layout style of the paragraph. The second column of options contains alignment settings for all text in the paragraph.



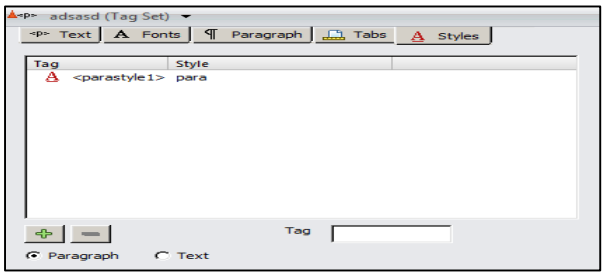
➢ Tag Set Properties - Tabs Tab

Tagged text can be positioned to specific tab positions. We define the tags for up to twelve tab stops (in left to right order).



➢ Tag Set Properties - Styles Tab

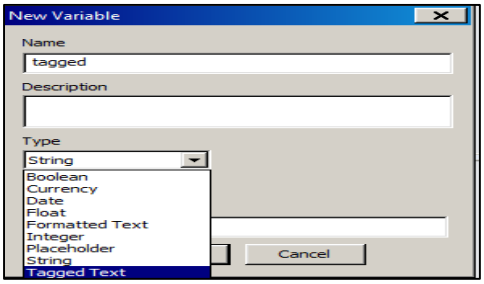
We can apply styles to tags, provided we specify the style on a style sheet and we assign the style sheet to the design page.



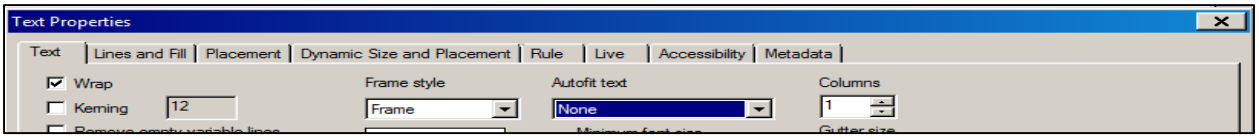
For more details on how to create Styles and Style Sheets, please refer to the STYLE and STYLE SHEETS section.

➢ Tagged text variables :

Now that we know how to create a tag set, for Exstream to scan tag sets in data we have to map the variable as a **tagged text.**



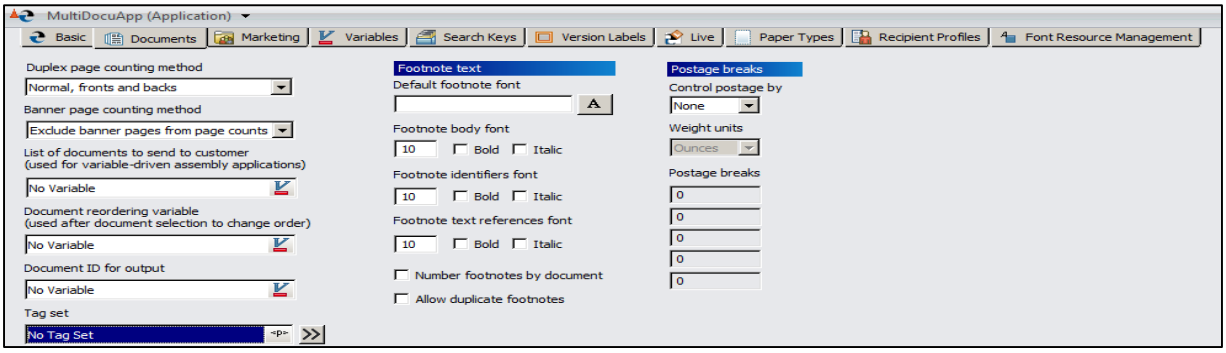
NOTE: *In Designer, place Tag Set variables only in text boxes that have the Autofit text dropdown list (on the Text Box tab of the Text Box Properties) set to None. The other Autofit options do not support Tag Sets.*



➢ Applications and Tag Sets:

Once we are done with creating tags sets, we need to add them to the application in order to configure our application with a specific tag set.

In the application properties inside document tab define the tag set in the tag set box.



2.10.1.9 Templates

There are two types of templates used in HP Exstream:

- **Page templates**: The details about Page templates is already discussed in Chapter 2 under the topic 2.1.4 Paper Types and Pages. Please refer to this chapter.
- **Message templates**: Message templates are used in the process of creating a message. For more information on how to create a message template, please refer to the Campaigns and messages section.

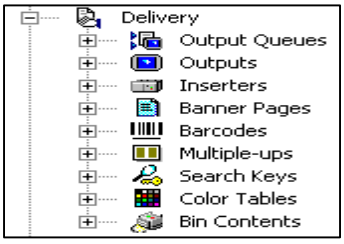
2.10.2 DELIVERY ENVIRONMENT

The delivery tab provides features to manage output devices and post processing equipment.

Some key features offered are:

- Control bins in inserters.
- Configure barcodes.
- Produce banner pages.
- Use imposition layout in booklets from multiple-up sheets.
- Create Search Keys.
- Override specific color specifications.
- Use System Queue Variables to control output.

Objects under Delivery



2.10.2.1 Output Queues

The details about Output Queue is already discussed in Chapter 2 under the topic 2.3 Output Queues. Please refer to this chapter.

2.10.2.2 Outputs

The details about Outputs is already discussed in Chapter 2 under the topic 2.3.6 Different types of Output Driver. Please refer to this chapter.

2.10.2.3 Inserters

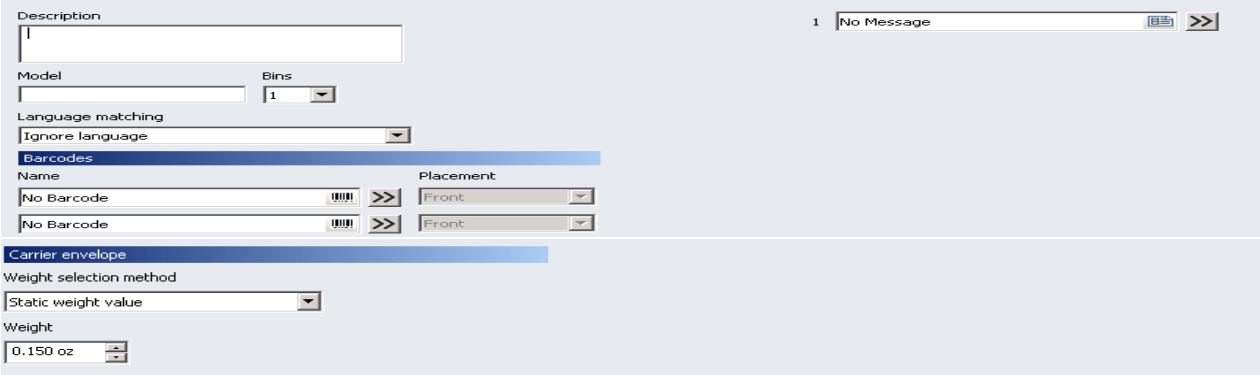
We use an Inserter object to define:

- A name (and, optionally, a thumbnail) of the pre-printed contents of each bin. The contents are referred to as insert messages.
- A barcode used to control insert selection.
- How customer language should be factored into the bin selection process.
- The weight of the envelope for the customer documents and inserts.

2.10.2.3.1 How to create an Inserter Object:

- Under **Environment**, expand the **Delivery** heading.
- Right-click the **Inserters** heading. Select **New Inserter** from the shortcut menu.
- The **Name** dialog box displays. Type a Name and, optionally, a **Description**.
- Click **Finish**.

Inserter in Property Panel



2.10.2.3.2 Setting Properties:

- Optionally, type the **model** of the inserter in the Model text box.
- Identify the number of bins available on the inserter in the Insert Stations drop-down list. The number we select here controls the number of bins displayed on the right side of the Property Panel. We can define up to 16 inserter bins.
- Select one of the following from the Language Processing drop-down list:
  - Ignore language to send the inserts listed in the bins no matter what the language of the customer.
  - Send inserts only if language matches customer to send the inserts listed in the bins if the language of the insert matches the language of the customer.
- Barcodes Area
  - Select the barcodes for this inserter configuration from the Barcode drop-down list. We can define up to three barcodes for a single inserter configuration.
  - Use the drop-down list below the barcode to tell Exstream whether to look on the Front, Back, or Both sides of the page for each barcode. Other options appear for use with

Multiple-Up sheets, based on the sequence number (or odd/even counts) of Multiple-up frames.

- ❖ To view the properties of the barcode we selected, click the adjacent button. To return, right-click a blank area of the Property Panel and select the inserters name from the shortcut menu.

- ➢ Carrier Envelope

  Use the Weight box to specify the weight of the envelope. Select either Ounces or Grams from the Units drop-down list to specify the unit of measure for the weight.

2.10.2.3.3 Adding inserters to Application:

In Output Queue properties, Inserter tab, we can reference one or more Inserter objects. If we list multiple inserters, we identify a particular inserter at run-time with the –SETQUEUEINSERTER Engine switch. (If we do not use this switch, the Engine selects the first inserter listed in this tab.)

2.10.2.4 Banner Page

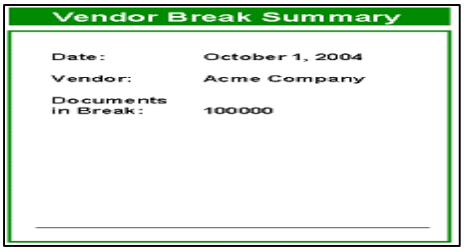
A banner page is an optional page added to the Output Queue to print at the start or at the end of a processing event. It can contain variable information such as Vendor ID or number of pages in the job, as an aid to splitting and sorting the finished output. Before we can associate a banner page to an Output Queue, we must create its design as a Page object in the Library.

2.10.2.4.1 Objects Needed to Create a Banner Page:

Before we can define a banner page object, we must have configured it in the Library:

- ➢ A suitable Paper Type.
- ➢ A Page object. We design a page specifically for the information we want to appear on the banner page.
- ➢ Exstream does not generate barcode objects on Banner pages.
- ➢ Banner pages cannot flow.

Sample Composed Banner Page



2.10.2.4.2 How to Create the Banner Page:

After we have designed a Page object, designate it as a banner page.

- ➢ Locate the **Banner** Pages heading in the Library under Delivery. Right-click the heading to access the shortcut menu.
- ➢ Select **New Banner**. The **Create a New Banner** Page dialog box opens.
- ➢ Type a **Name** and a **Description**. Click the button. The Select Page dialog box opens.
- ➢ Highlight the Page object we created in Designer to use as a banner page and click OK.
- ➢ The page name appears in the Page text box. Click **OK**.

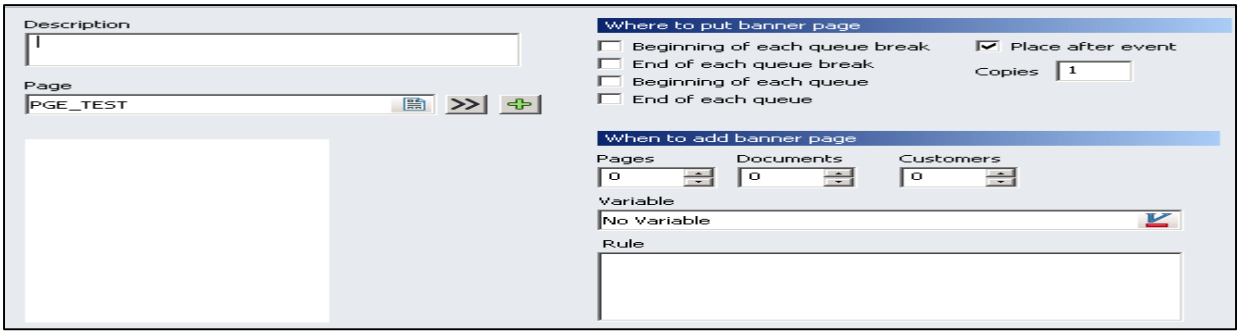
The new banner page opens in the Property Panel for us to define.

2.10.2.4.3 Properties of the Banner Page:

➢ Usage Area

The majority of the properties for a banner page have to do with selecting when the banner page should print. In the Usage Area, we select which event causes the banner page to print.

When each Queue Break begins	When each Queue Break ends	When a new Output Queue begins processing
When an Output Queue ends processing	By number of pages	By number of documents
By number of customers	By Variable change	By Rule



➢ Page

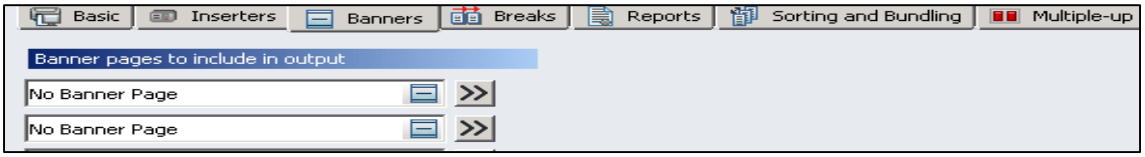
The Page box at the upper left displays the choice we made when we created the banner page. Below it, we see a thumbnail of the page.

1. To change the page, click the button in the Page text box. The Select Page dialog box opens.
2. Choose an existing page in the **Library** to use as the banner page. Click **OK**.

2.10.2.4.4 Associate the Banner Page to an Output Queue:

The last thing we need to do with our banner page is reference it in an Output Queue.

1. Drag to the Property Panel the **Output Queue** for the banner page.
2. Click on the **Banners** tab.
3. Select the **banner page** object from the drop-down list.



2.10.2.5 Barcodes

Please refer to Chapter 2 under the topic 2.9 Barcodes for more details.

2.10.2.6 Search Keys

Please refer to Chapter 2 under the topic 2.7 Search Keys for more details.

2.10.2.7 Color Tables

Color Tables enable us to customize the colors on output devices so that they coincide with design colors. Color Tables enables us to override the values of the named colors in the customized Color Families objects under Design.

CHAPTER 3 HP EXSTREAM DESIGN OVERVIEW

LEARNING OBJECTIVES: At the end of this chapter, the reader would be able to understand

- *Text Box and its Properties*
- *Different types of Tables and its properties*
- *Designing Charts*
- *Different types of Flow objects*

3.1 TEXT BOXES

3.1.1 Introduction to Text Boxes

Text Boxes are the design elements that act as a container for text in the designer. It is used to place text on graphic messages and pages. They can also be embedded inline on objects such as text messages.

Benefits

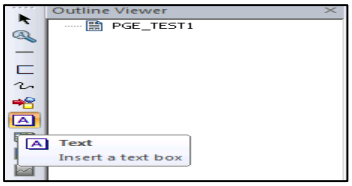
Generally we use a text box for any of the following purposes:

- Static or dynamic content holder
- Container for Importing TIFFs/RTFs/Logos

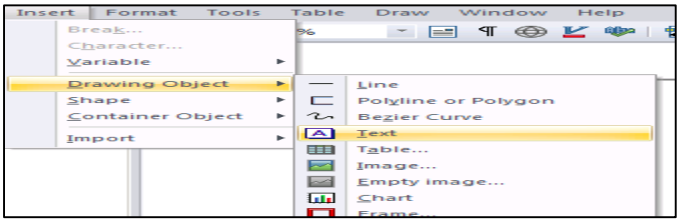
3.1.2 Creating Text Boxes

There are two ways to create Text Boxes in Design Manager:

- On the Drawing Objects Toolbar, click on .



- On **Insert** Menu Click on **Drawing Object**, then **Text.**



3.1.3 Sizing Text Boxes

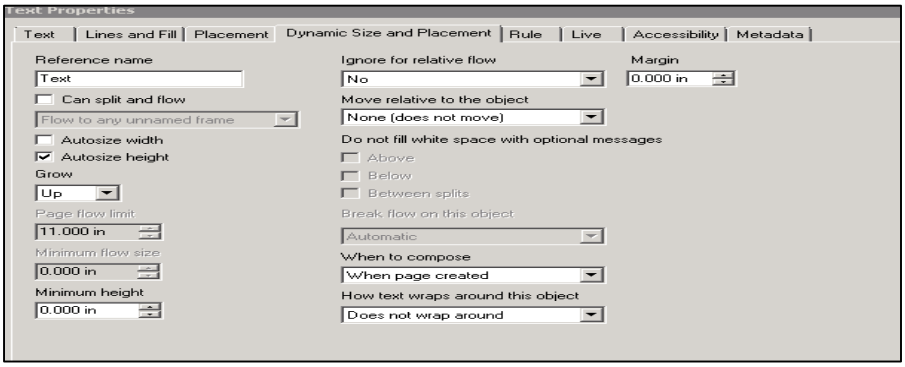
Text Boxes can be sized in two different ways:

1. **Autosizing Text Boxes:**

Autosizing enables the text box to grow larger or smaller depending on the text entered. This feature is necessary when the size of the text box is not fixed and can grow/shrink during the designing and processing stages. To set this option

- Move to the **Dynamic Size and Placement** tab of text box properties.
- Check the Autosize height/ Autosize width option.

To turn off we have to uncheck these options.

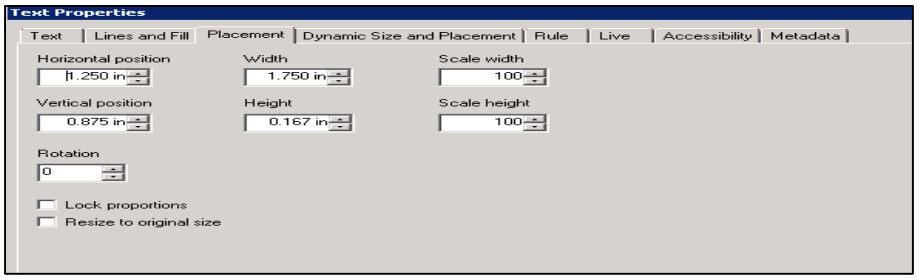


Vertical growth:

When the autosize height option is activated, the **Grow** drop-down list is also activated. This drop-down list allows the text box to either **Grow Up** or **Down**. A text box that Grow **Up** cannot split to another page.

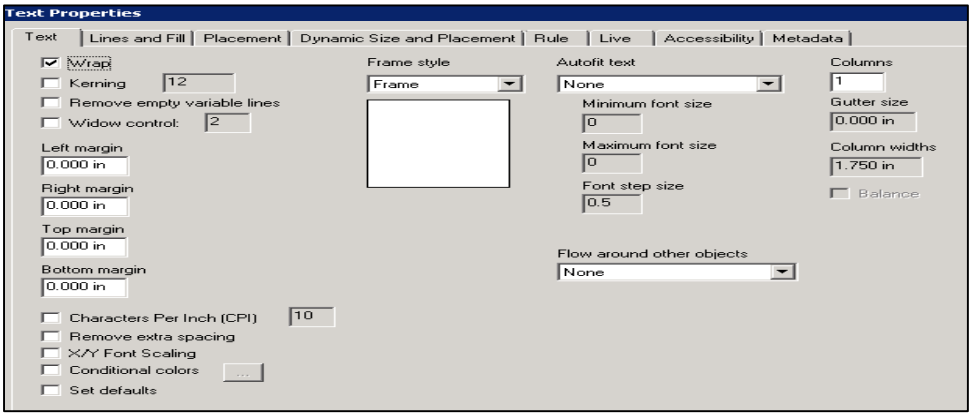
2. Manually sizing a text box:

When the text box contains static text with fixed size then the size of the Text Box is set manually. We can change the height or width of the text box by dragging the sides of the text box or selecting the **Placement** of **Text Box** Properties dialog box.



3.1.4 Tabs in Text Boxes

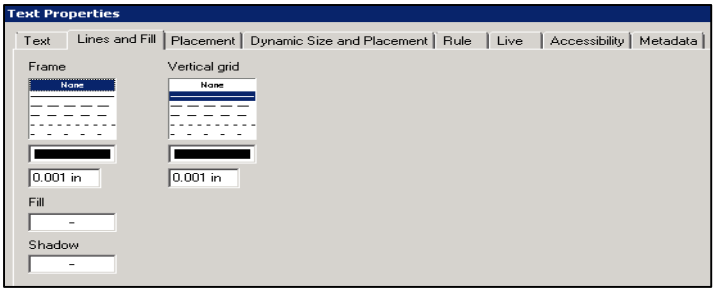
- **Text Tab**



This tab sets the generalized properties of the text. Explained below are some of the important properties.

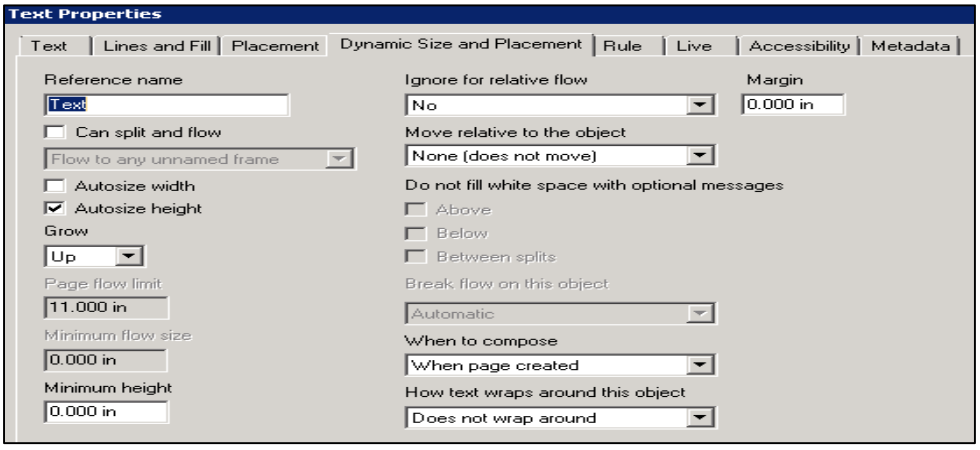
- **Wrap:** This checkbox is selected when we want to wrap the text to the next line.
- **Kerning:** Kerning moves characters closer together or farther apart to adjust the spacing between letters. When we select this checkbox we need to specify a size to allow kerning for characters at that point size and greater.
- **Remove empty variable lines:** This checkbox is used to remove blank lines cause by variables with no value.
- **Widow control:** This checkbox lets us control the number of lines of text that must be together when the text box splits. After selecting this checkbox we need to specify the count of lines in its adjacent box.
- **Margins:** We can set the margins for the text box for **Left**, **Right**, **Top** and **Bottom** positions.
- **Frame style:** This drop-down list is used to place lines or a shape in the text box, behind the text.

- **Lines and Fill Tab:**



Here we select the fill-color and line style for the frame of the text box.

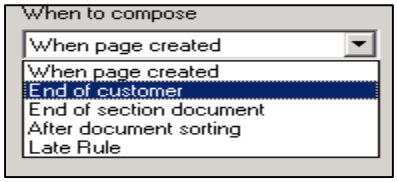
- **Placement Tab:** This tab has already been discussed above.
- **Dynamic Size and Placement:**



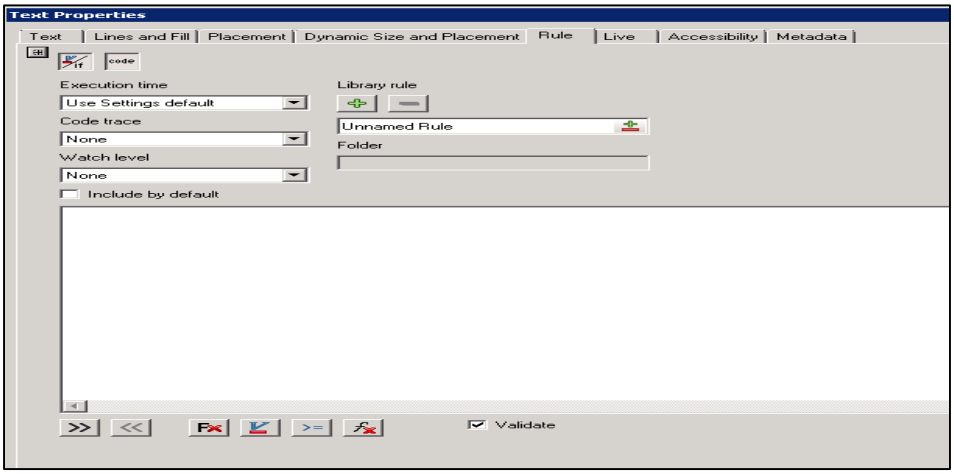
Explained below are some of the important properties.

- **Can split and flow:** This option allows the text box to be split across pages.

- **Autosize width, Autosize height and Grow:** Already discussed.
- **Page flow limit:** This option sets the page limit for flowing. When the text box exceeds the limit, the content will flow to a new page.
- **Move Relative to the object:** This option will move the text box relative to an object **above/left/right**.
- **When to compose:** One of the following options is chosen for identifying when to compose the text box:



- **Rule Tab:**



We can add rules to the text box via this tab. We can use the **Code** panel to create a rule.

3.2 TABLES

3.2.1 Introduction

Tables are objects created exclusively in Designer that consist of cells arranged in rows and columns. Tables can contain repeating rows and columns, headers and footers, and static rows. Rows and columns can be controlled by inclusion rules. Tables can grow, split, and flow to subsequent pages in the design.

3.2.2 Table Types

The following types of tables are available in Exstream Designer:

Type	Symbol	Description
Simple Table		Simple Tables are the most basic tables and the only table we can create without the Advanced Tables module.
Basic Automated Table		Basic Automated Tables lets us create a table with rows, headers, and footers that automatically grow to accept new data.
Automated Table with Sections		Automated Tables with Sections lets us create sections in our table. We can also enable section data processing for the table.
Automated Table with Levels		Automated Tables with Levels lets us create levels of sections within your table. We can assign a numeric level to a section in a table.
Basic Automated Table with Automated Columns		Basic Automated Tables with Automated Columns lets us create a Basic Automated Table with the added ability to create Automated columns.
User Table		User Tables lets us create a table capable of using all offered table features.

Benefits

- Table lets us communicate information in a clear, concise manner.
- We can use tables to summarize transactions, condense lengthy information, and compute values.

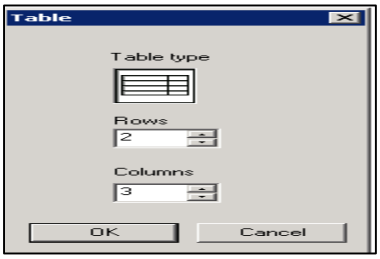
3.2.3 Creating and Defining Tables

A table lets us organize and present information using columns and rows. We can add a table to a page, paragraph, or message in Designer.

3.2.3.1 Creating Tables

To create a table in the Design Manager:

1. In the Drawing Tools Toolbar we have to click the symbol  .The **Table** Dialog box opens.



2. Click on the **Table Type** icon and then select the radio button next to the table type we want to create.
3. In order to use sectioned data in the table, we need to select the Enable data section processing check box.
4. Click **OK**. The **Table Type** dialog closes.
5. In the **Rows** box, we need to enter the number of rows we want for the table.
6. In the **Columns** box, we need to enter the number of columns we want for the table.

NOTE: *The Enable data section processing check box is active only if the table type is Automated Table with Sections, Automated Table with levels, or User Table.*

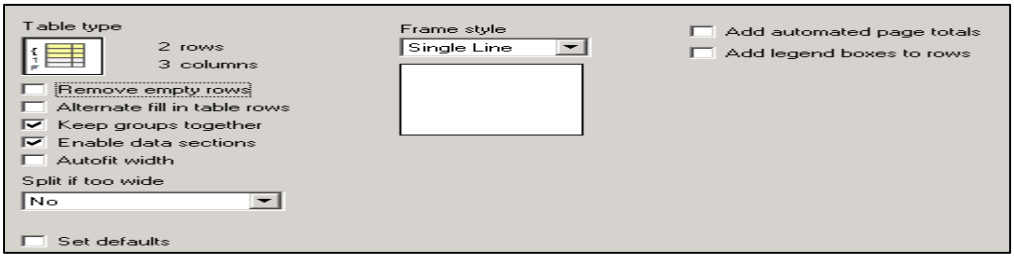
3.2.3.2 Table Properties

After creation of a table, we have to define its properties. Options available varies depending on the table type we have selected. We define the table properties from the Table Properties dialog box. A table's properties can be defined by any of the following methods:

1. Right-click the table and select Table Properties.
2. Click above the table. is available only when the table is active.
3. From the Menu bar, select **Tables > Table properties**. The Tables menu is available only when the table is active.

3.2.3.3 Defining Table

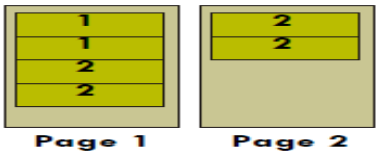
- **Table Tab:**



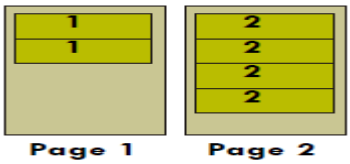
- **Table Type:** The Table type icon shows the type of table we are creating.
- **Rows and Columns*:*** The number of rows and columns in the table appears on the Tables tab.
- **Remove Empty Rows:** The Remove empty rows check box lets us control the use of empty rows in a table. If we select this check box, rows not containing text or other objects do not appear in the output. If we clear this check box, all table rows appear in the output, even if they are empty.
- **Alternate Fill in Table Rows:** This property allows us to apply a background color to every other row. We must also apply the background color we want to use to the cells in the first row of our table. The engine applies color to the first row and continues to alternate the fill through additional rows. If we clear the Alternate fill in table rows check box, all rows have the same background color.

NOTE: *The Alternate fill in table rows check box is not available for Simple Tables and tables that use the serpentine multi-column flow method.*

- **Keep Groups Together:** If we select this check box, repeating rows will be kept together when the table splits across pages. If we clear this check box, repeating rows can separate when the table splits.



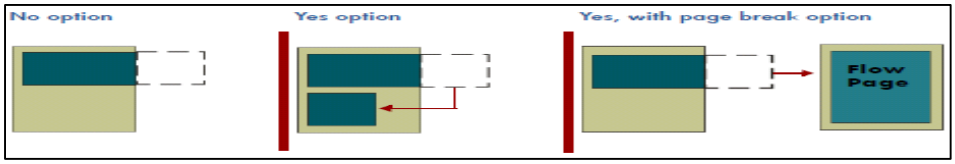
Keep groups together check box cleared



Keep groups together check box selected

NOTE: *In the final produced output, the Keep groups together setting overrides the Widow.*

- **Enable Data Sections*:*** The Enable data sections check box tells the Designer that the table can be controlled by sectioned data in the data file. This check box gets selected if we had selected the **Enable data section** processing on the Table Type dialog box.
- **Autofit Width:** The **Autofit width** check box allows the column widths to adjust to fit content during the engine run. In the Target width box, we have to specify the total width of the table. This measurement is the width of the table after columns have been excluded by rules or included because of automated columns.
- **Split if Too Wide:** The Split if too wide drop-down list lets us control what happens to the table if it grows wider than the page. The following options are available:
    1. **No**: Does not split. Any information beyond the width of the page is lost.
    2. **Yes**: Splits and the overflow is placed directly below the first section of the table. Specify the amount of space between portions of the table in the adjacent box.
    3. **Yes, with page break**: Splits and the overflow is placed on another page containing an overflow frame.

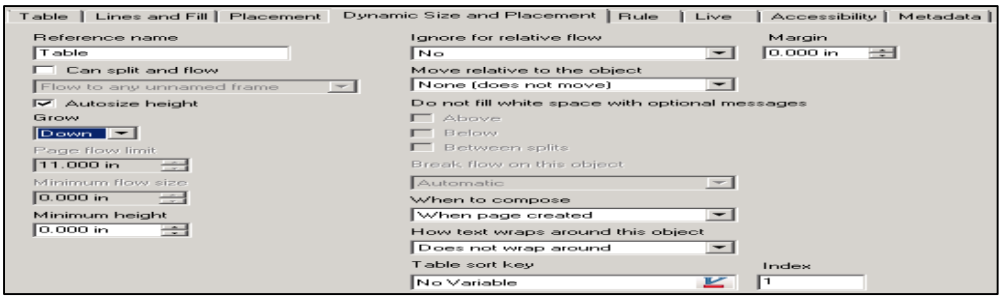


NOTE: *We must select Can split and flow on the Dynamic Size and Placement tab if we specify Yes or Yes, with page break from the Split if too wide drop-down list. If we do not select Can split and flow, we will receive a message when we exit the Table Properties dialog box and automatically selects the option.*

- **Frame Style:** The Frame style drop-down list lets us select an outline style for how the table appears. We can Select one of the following options:
    - **Single Line**: The table is outlined with a single line. This is the default.
    - **Double Line**: The table is outlined with a double line.
    - **Rounded**: The corners of the table are rounded.

➢ **Dynamic Size and Placement Tab:**

This tab helps us to define how a table grows on a page.



❖ **Grow:** Tables by default are set to grow downward. Simple tables, unlike other objects and tables, can be set to grow upward. This option is useful to prevent a static object, which must remain at the bottom of the page, from being overlapped by a growing table. For example, if a page has a remittance slip or a coupon on the bottom, we can place a simple table (or text box) above it that can grow upward.

NOTE: *Nothing can be relative to objects set to grow upward. In addition, objects set to grow upward cannot be relative to other objects. Objects that grow upward cannot flow.*

To make a Table grow upward we have to follow the below steps:

1. Uncheck the Can split and flow check box.
2. Select the Autosize height check box.
3. From the Grow drop-down list, we have to select **Up**.
4. From the Ignore for relative flow drop-down list, we have to select **Yes**.
5. Click OK.

3.2.4 Creating and Defining Columns

3.2.4.1 Creating Columns

Columns are the vertical structures in tables that we define to meet our needs. We can insert a column into any existing table. To insert a new column, select a column and we can do one of the following:

1. Right-click a selected column and select **Insert column before** or **Insert column after**.
2. Select **Insert column before** or **Insert column after** from the **Table** menu on the Menu bar.
3. Click the ◻ above the column and select **Copy**.

➢ **Copying:**

1. Right-click a selected column and select **Copy column**.
2. Select **Table > Duplicate Column** on the Menu bar.
3. Click the ◻ above the column and select **Copy**.

➢ **Deleting:**

To delete a column, select a column and we can do one of the following:

1. Right-click a selected column and select **Copy column**.
2. Select **Table > Duplicate Column** on the Menu bar.
3. Click the ◻ above the column and select **Copy**.

➢ **Resizing:**

We can manually resize a column by dragging its border. We can use the Distribute columns function to automatically make the selected columns equal in size by any of the following ways:

1. Right-click a selected columns and select Distribute columns.
2. Select Table > Distribute columns on the Menu bar.

NOTE: *We cannot resize by dragging the top or bottom border of a table.*

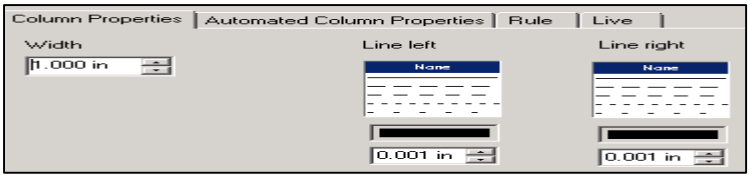
3.2.4.2 Defining Columns

To define column properties, select a column and do one of the following:

1. Right-click a selected column and select **Column Properties**.
2. Select **Table > Column properties** on the Menu bar.
3. Click above the column and select Properties.

➢ **Column Properties Tab:**

The **Column Properties** tab lets us control the visible formatting of the column. The options for columns vary depending on the table type.



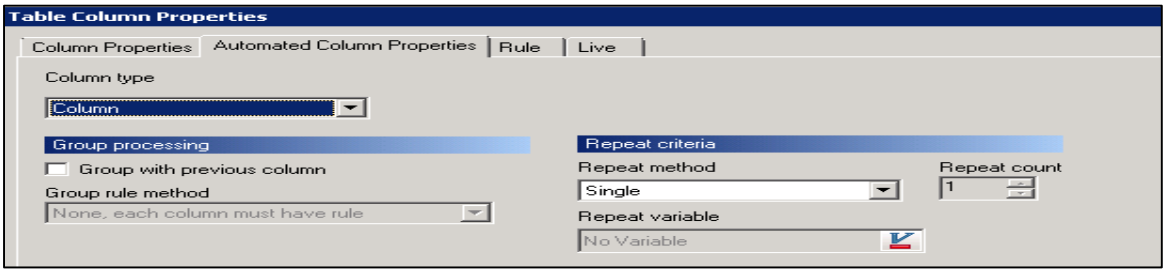
❖ **Width:** The Width box reflects the current width of the selected column. We can enter a measurement or use the arrow buttons to alter the width. We can also click a column border and drag it to change the column width. If we select multiple columns, all columns will adjust to the width selected.

❖ **Line Left and Line Right*:*** The Line left and Line right areas lets us format the lines that appear on each side of the column. We need to select the style for the column line in the box. None is the default. We can also specify the color and thickness of the column lines.

NOTE: *The Autofit width drop-down list is active only if we select the Autofit width check box on the Table tab in the Table Properties dialog box. In addition, we also specify the total width of the composed table in the Table Properties dialog box.*

➢ **Automated Column Properties Tab:**

The Automated Column Properties tab controls the repetition of automated columns. This tab is available only if we are using a Basic Automated Table with Automated Columns or a User Table.



- **Column Type:** We can select one of the following column types for the current column:
  1. **Column**: An automated column that can be set to repeat.
  2. **Header**: A header that appears only once.
  3. **Repeating header**: A header that appears each time the table flows to another page.
  4. **Repeating header except first**: A header that appears each time the table flows to another page, but does not appear on the first page.
  5. **Footer**: A footer that appears only once.
  6. **Repeating footer**: A footer that appears each time the table flows to another page.
  7. **Repeating footer except last**: A footer that appears each time the table flows to another page, but does not appear on the last page.
- **Group Processing Area*:*** This area is active only if we select Column from the Column type drop-down list.

  Group with Previous Column: This check box groups the selected column with the one before it.

  Group Rule Method: This drop-down list is active only if the selected column is the first column in a group. We can select one of the following options:

  1. **None, each column must have rule**: Each column is controlled by its own rule. This is the default setting.
  2. **Rule controls all columns in group**: A single rule controls the entire group. If the conditions are met, the entire group is included. The rule must be applied to the first column in the group or it is ignored.
- **Repeat Criteria Area:** The Repeat Criteria area lets us control if and how a column repeats in the table.

  Repeat Method:
  The Repeat method drop-down list lets you specify how Designer determines the number of columns in a table. The following options are available:

  1. **Single:** The column appears once. This is the default.
  2. **Fixed number:** The column appears the number of times we specify in the Repeat count box.
  3. **Number of elements in array:** The column repeats based on the number of elements in the array variable we specify in the Repeat variable box.
  4. **Value of variable:** The column repeats based on the value of the variable we specify in the Repeat variable box.

3.2.5 Inserting and Defining Rows

3.2.5.1 Inserting Rows

We can insert a row into any existing table. To insert a new row, select an existing row in the table and do one of the following:

1. Right-click a selected row, and from the shortcut menu, select **Insert row before or Insert row after**.
2. From the Menu bar, select **Table > Insert row before or Insert row after**.
3. To the right of the row, click ▫ and select Insert before or Insert after.

➢ **Copying:**

1. Right-click a selected row, and from the shortcut menu, select Duplicate row.
2. Select **Table > Duplicate rows** on the Menu bar.
3. To the right of the row, click ▫ and from the shortcut menu, select Duplicate.

➢ **Deleting:**

1. Right-click a selected row, and from the shortcut menu, select Delete row.
2. Select **Table > Delete Rows/Columns** on the Menu bar.
3. To the right of the row, click ▫ and from the shortcut menu, select Delete.

➢ **Resizing:**

We can manually resize a row by dragging its border. We can use the Distribute rows function to automatically make the selected rows equal in size by any of the following ways:

1. Right-click a selected row, and from the shortcut menu, select **Distribute Rows**.
2. On the Menu bar, select **Tables > Distribute Rows**.

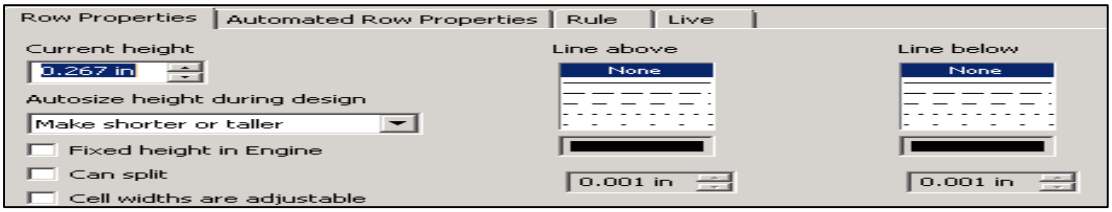
3.2.5.2 Defining Row Properties

To define row properties, select a row and then do one of the following:

1. Right-click a selected row, and from the shortcut menu, select Row Properties.
2. From the Menu bar, select Table > Row properties.
3. To the right of the row, click ▫ and from the shortcut menu, select Properties.

➢ **Row Properties Tab:**

This tab lets us control the formatting of the row. The options available for rows varies depending on the table type.



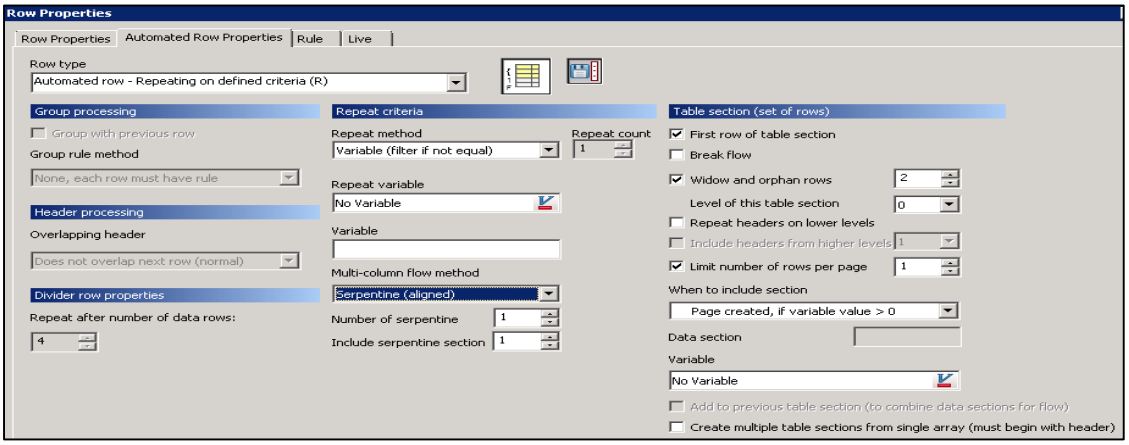
- **Current Height:** The Current height box shows the current height of the selected row. We can enter a measurement or use the arrow buttons to alter the height.
- **Autosize height During Design:** We can select one of the following methods the engine applies to adjust the rows during processing:

1. **Make shorter or taller**: The row can grow or shrink vertically based on content. This is the default.
2. **Shorter only**: The row can shrink vertically if there is not enough text to fill the row. The current height is the tallest the row can be.
3. **Taller only**: The row can grow vertically if the text is too large for the row. The current height is the smallest the row can be.
4. **Do not resize**: The row remains the same size, regardless of content.

- **Fixed Height in Engine:** To keep a row at its current height during composition, we need to select the **Fixed height in Engine** check box.
- **Can split:** To enable the row to split, we need to select the Can split check box.
- **Cell Widths are Adjustable:** To change the width of a single cell in a row without changing the width of the row, we need to select the **Cell widths are adjustable** check box.
- **Line Above and Line Below:** The **Line above** and **Line below** areas lets us format the lines that appear above or below a row.

Automated Row Properties Tab:

The **Automated Row Properties** tab controls the repetition of the rows.



- **Row Type:** It specifies the row type of the current row. The below options are:
  1. **Automated, based on repeat criteria:** Create a data row that repeats, based on repeat criteria.
  2. **Not automated:** Creates a static row.
  3. **Repeating divider:** Create a divider row that repeats after groups of data rows.
  4. **Header:** Create a header row that appears only once, at the start of the table.
  5. **Repeating header:** Create a header row that appears each time the table flows to another page.
  6. **Header, if at top of flow frame:** Create a header row that appears only if the table is located at the top of a flow frame.
  7. **Header, if NOT at top of flow frame:** Create a header row that appears only if the table is not located at the top of a flow frame.

8. **Header, if 1st occurrence on page (Used with table sections only):** Create a header row that appears only when sections are applied to the table and it is not the first occurrence of the section on the page.
9. **Header, if NOT 1st occurrence on page (Used with table sections only):** Create a header row that appears only when sections are applied to the table and it is not the first occurrence of the section on the page.
10. **Footer:** Create a footer row that appears only once, at the end of the table.
11. **Repeating footer:** Create a footer row that appears each time the table flows to another page.
12. **Repeating footer, except last:** Create a footer row that appears each time the table flows to another page but does not appear on the last page.
13. **Footer: add after all lower-level row sets:** Create a footer row that appears after each subsection.

- **Group Processing Area:** The options in the **Group Processing** area lets us control how rows are grouped.

  Group with Previous Row: Select the **Group with previous row** check box if we want to group a row with the one before it. Each time the first row appears, the row grouped with it also appears.

  Group Rule Method: The Group rule method drop-down list lets us specify the rules method to control the group or individual rows of the group. The Group rule method drop-down list is active only if the selected row is the first row in a group. We can select one of the following options:

  1. **None, each row must have own rule**: Each row is controlled by its own rule. This is the default.
  2. **Rule controls all rows**: A single rule controls the entire group.
  3. **Rule selects one or more group rows**: A single rule selects individual rows from within the group based on the criteria that are met. Up to four rows can be included for each condition. The rule must be applied to the first row in the group or it is ignored.

- **Repeat Criteria Area:** The **Repeating row** area lets us define the repeat criteria of an automated row.
  Repeat Method: we can select one of the below repeating methods:

  1. **Fixed number**: The row repeats based on the number we specify in the Repeat count box.
  2. **Number of elements in array**: The row repeats based on the number of elements in the array variable we specify in the Repeat variable box.
  3. **Variable (filter if not equal)**: The row repeats based on the number of elements in the array variable we specify in the Repeat variable box. Excludes any variable values not found in the Variable filter box.
  4. **Variable (filter if equal)**: The row repeats based on the number of elements in the array variable in the Repeat variable box. Excludes any variable values found in the Variable filter box.

5. **Value of variable**: The row repeats based on the value of the variable we specify in the Repeat variable box.

- **Multi-Column Flow Method:** The Multi-column flow method drop-down list is active if we select any option other than fixed number from the Repeat method drop-down list. The Multi-column flow method drop-down list lets us specify how variables fill table cells. The below options are available:

  1. **Fill columns first:** Variables fill cells vertically before continuing to the next column.
  2. **Fill rows first:** Variables fill cells horizontally before continuing to the next row.
  3. **Duplicate entries:** Variables in a column repeat in each cell for the row.
  4. **Serpentine (aligned):** Cells are placed next to each other with no space between cells.
  5. **Serpentine (spread):** Cells are evenly distributed across the row with spaces between cells so the row can fill equally.

- **Table Section (Set of Rows) Area:** The Table section area is available only if we use an Automated Table with Sections, Automated Table with Levels, or User Table type.

- **First Row of Table Section:** Select the First row of table section check box to create sections within a table. When we select this check box, the remaining options in the Table Section (set of rows) area become active.

- **Break Flow:** Select the Break flow check box to allow the table to break when the engine encounters the selected row. The table flows to the next available flow frame.

- **Widow and Orphan Rows:** HP Exstream does not support the Widow and orphan rows setting if the selected rows contain a header row or the first row of a table section. Select the Widow and orphan rows check box to prevent rows from being isolated on pages. Specify the minimum number of rows per page for this section in the adjacent box. When a table flows to another page, the number of rows placed on the overflow page must be greater than or equal to the number specified in the box. If the number of rows is less than the number specified, the entire section is placed on the overflow page.

- **Level of This Table Section:** If we want to nest sections within other sections, we can use the **Level of this table section** drop-down list to select a number to designate the level of the current section. The highest level in the **Level of this table section** drop-down list is **1** and the lowest level is **20**.

NOTE: *The Level of this table section drop-down list is available only if we are using an automated table with levels, or a User Table.*

- **When to Include Section:** We need to select an option specifying when the current section is included in the table.

1. **When page created**: The section is always included in this table
2. **Page created, if variable count > 0**: The engine includes the section when the variable count is greater than zero and if the variable is present in the data. Specify the variable in the **Variable** box below.
3. **Page created, if variable value > 0**: The engine includes the section when the value of the specified variable is greater than zero in the customer data. Specify the variable in the **Variable** box below.
4. **Page created, if variable count = 0**: The engine includes the section when the variable count is zero.
5. **Page created, if variable value = 0**: The engine includes the section when the value of the specified variable is equal to zero in the customer data.
6. **Named data section, each**: The engine includes the section when the data section appears in the customer data. Enter the name of the section in the Data section box below.
7. **Data section, first only**: The engine includes the section when the first occurrence of the data section is encountered. Enter the name of the section in the Data section box below.
8. **With table section at next level, each**: The engine includes the section each time a section follows the current section.
9. **Table section at next level, first only**: The engine includes the section the first time a section follows the current section.
10. **After all data sections (end of customer)**: The engine includes the section after all section data is completed for the customer.

- **Create Multiple Table Sections From Single Array:** To create the look of sectioned data with simple data, we need to select the **Create multiple table sections from single array (must begin with header)** check box. The table must be set to split and flow for this feature to work properly. We cannot nest a single array section within another single array section.

- **Row Breaks Area:** The Row breaks area lets us specify how and when a row will break. The **Break Methods** available are:
  1. **Fixed Number:** Breaks after a specified number of rows. Specify the number of rows in the **Number per break** box.
  2. **When Variable Value Changes:** Breaks when the value of the variable is different from the previous value. Specify the variable in the Variable box.
  3. **When Variable Does not Equal:** Breaks when the value of the variable does not equal the specified value. Specify the variable in the Variable box and the value in the Variable filter box.
  4. **When Variable Equals:** Breaks when the value of the variable equals the specified value. Specify the variable in the Variable box and the value in the Variable filter box.
  5. **Value of Variable:** Breaks after the engine places all the array elements of the specified variable in the table. Specify the variable to control the break in the **Variable** box.

3.2.6 Defining Cells

3.2.6.1 Joining and Splitting Cells

We can combine cells in a row. To join cells, we have to select the cells we want to join.

1. Right-click and select **Join cells**.
2. Select **Table** > **Join cells** on the Menu bar.

If we join cells that contain content, only the content of the cell farthest to the left is kept in the joined cell.

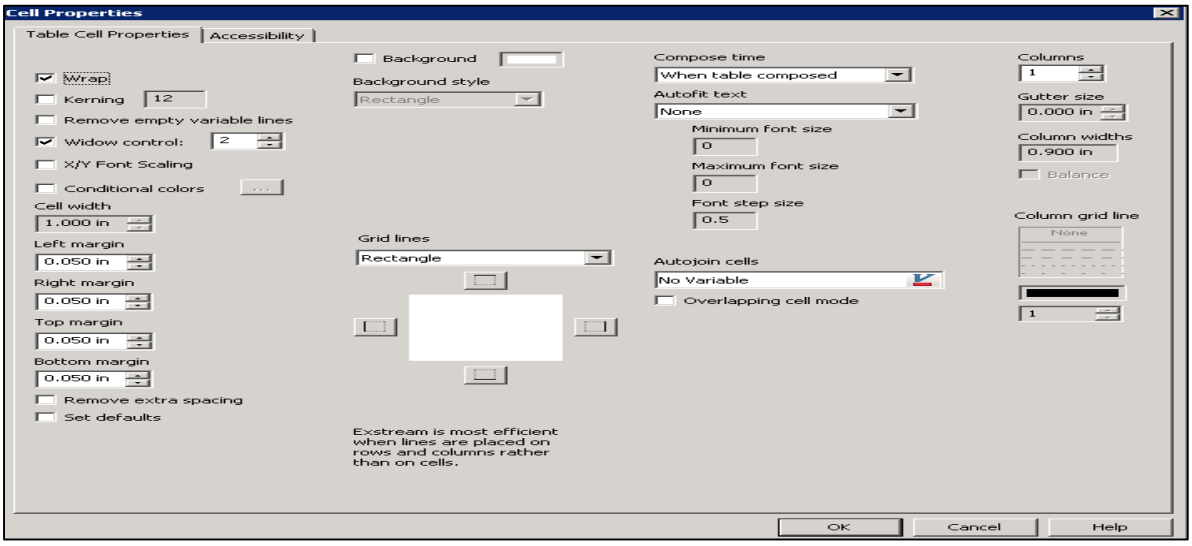
To separate joined cells, select the joined cell and do one of the following:

1. Right-click a selected joined cell and select **Split cells**.
2. Select **Table** > **Split cells** on the Menu bar.

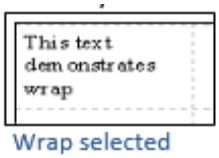
3.2.6.2 Table Cell Properties

To define cell properties, we need to select the cells we want to define and do one of the following:

1. Right-click a selected column and select Cell Properties.
2. Select Table > Cell properties on the Menu bar.
3. Triple-click a cell.



- **Wrap:** Select the **Wrap** check box to continue text on the next line in the cell. The Wrap check box is selected by default.

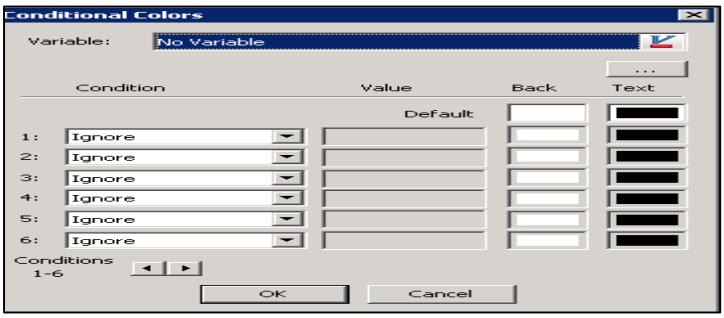


Wrap selected

Wrap cleared

- **Kerning:** The **Kerning** check box lets us move characters closer together or farther apart when the letter spacing is not visually correct. Any font that is larger than the font size we enter (must be larger than six points) in the **Kerning** box is kerned.

- **Remove Empty Variable Lines:** The **Remove empty variable lines** check box lets us remove blank lines caused by variables with no values.
- **Widow Control:** The **Widow control** check box lets us adjust the widow/orphan control on the text if cell splits.
- **X/Y Font Scaling:** The **X/Y Font Scaling** check box lets Designer resize fixed-size fonts to fit in the cell.
- **Conditional Colors:** Conditional colors are used to apply different colors to the cell and text, depending on variable information. To use conditional colors, we need to select the Conditional Colors check box.



  - **Variable:** Click to access the Variable panel dialog box where we select the Variable on which to base the condition.
  - **Condition:** We can set up to 30 different conditions and colors for the variable we select. Below are the conditions:
    1. **Ignore**: The color is applied to the cell regardless of the Variable value. This is the default.
    2. **Equals**: The value of the Variable must match Value exactly.
    3. **Not Equal**: The value of the Variable must be anything other than Value.
    4. **Greater Than**: The value of the Variable must be more than Value.
    5. **Less Than**: The value of the Variable must be less than Value.
    6. **Greater or Equal**: The value of the Variable must be Value or more.
    7. **Less Than or Equal**: The value of the Variable must be Value or less.
    8. **Greater than Zero**: The value of the Variable must be more than zero.
    9. **Less than Zero**: The value of the Variable must be less than zero.
    10. **Equal Zero**: The value of the Variable must be zero.
  - **Value:** Enter a constant value in the Value box to compare against the variable.
  - **Background and Text Colors:** If the condition is true, the colors of the background and text change to the colors specified in the **Back** and **Text** color wells. If the condition is false, the colors in the **Default** color wells are the cell colors.
- **Margins:** We can specify text margins on all four sides of the cell. The default size depends on the unit measurement.
- **Remove Extra Spacing: Remove extra spacing** check box is used to remove any extra spaces before the first paragraph and after the last paragraph of text within a cell.
- **Background:** We select the **Background** check box to specify the color and shape of the cell. Click the color well to select the color for the cell background.
- **Background Style:** We can select this check box to specify the style of the background.

- **Grid Lines:** The Grid lines area provides border line styles around the cell like rectangle, rounded etc.
- **Compose Time:** The **Compose time** drop-down list lets us specify when the contents of the cell are placed during a production run. Compose time is useful if the cell contains a formula that cannot be composed when the page is created because not all of the information is available at that time. We can select one of the following options:

  1. **When table composed**: Places cell contents when the table is composed. This is the default.
  2. **End of customer**: Places cell contents when the end of the customer is reached
  3. **End of section document:** Places cell contents when the end of the current section is reached. Can be used only if we are using section-driven documents.

- **Autofit Text:** The **Autofit text** drop-down list lets Designer resize fonts automatically to fit in the cell. If we use Autofit text, all the variable information fits in the cell, but the font size automatically varies with the size of the variable information. Select one of the following options:

  None: Does not automatically resize the text. If the text is too large to fit in the cell, it is truncated. This is the default setting.

  1. **Make smaller to fit**: Automatically resizes the text to the largest size that can fit in the cell, but no larger than the font size of the variable.
  2. **Maximum size that fits**: Automatically resizes the text to the largest size that can fit in the cell.
  3. **Both**: Uses both of the above options to dictate automatically resizing the text.

- **Autojoin Cells:** The Autojoin cells option lets us join cells that each contain a variable with values that match. Designer compares the values of the Autojoin variable in the cells. If the values match, the cells become one.

- **Overlapping Cell Mode:** The Overlapping cell mode check box lets the contents of a cell exceed its vertical boundaries. When the engine runs, the following actions occur:

  1. The overlapping cells are ignored to determine the height of the table. Objects that are relative to it ignore the overlapping cells.
  2. When a row with an overlapping cell is placed on the page, the overlapping cell must fit on the page or the row is not included on the page.
  3. Headers and footers can be overlapped at the bottom or top by growing text.

3.3 CHARTS

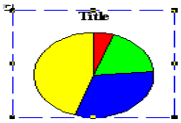
- Charts are visual representations detailing relationships between two sets of data or the relationship of a part to a whole.
- Charts are read more quickly than raw data.
- Purpose of Chart is to show and compare changes or relationships and to bring facts.

3.3.1 Types of charts available in Exstream

Pie chart	Line chart	Area chart	Progress bar chart
Bar chart	Stacked bar chart	Comparative bar chart	Horizontal bar chart
Horizontal stacked bar chart	Calendar chart	Range bar chart	Label Chart
Radar chart	Comparative Line chart	Scattergram chart	Floating Bar chart

3.3.2 Creating a chart

1. On the Drawing Tools toolbar, click the button. We can notice that our pointer changes which indicates that we are placing a chart on the page.
2. Click the page. A chart is inserted.

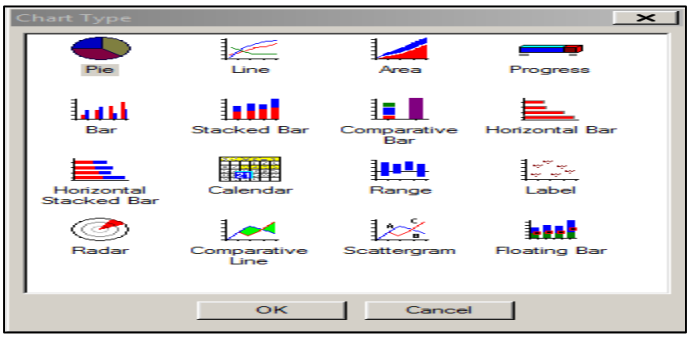


3. Drag the lower right corner to enlarge the chart to the approximate size our chart should occupy.

3.3.3 Define the Type of Chart

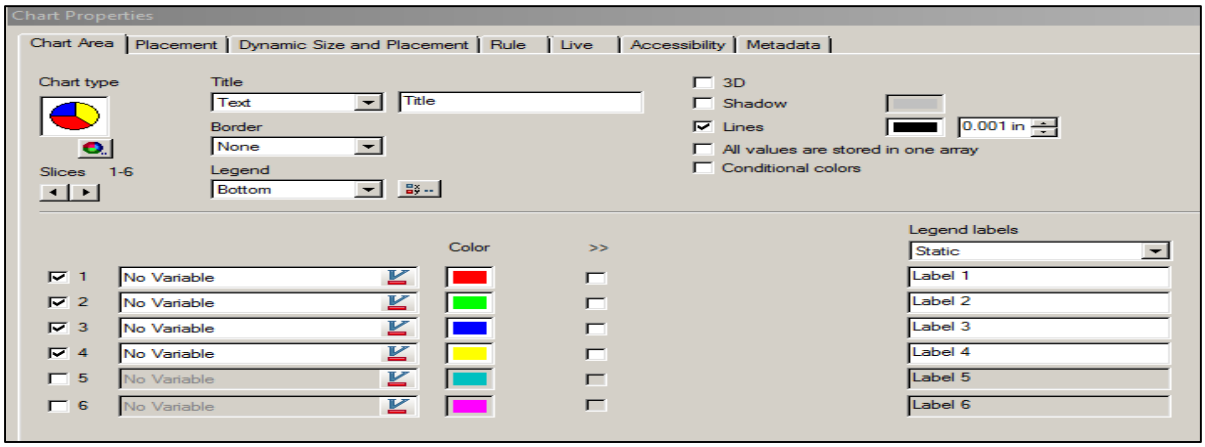
By default, when we place a chart on the page, a pie chart is placed. If we do not want to create a pie chart, the first thing we must do is change the chart type on the Chart tab.

1. Right-click the chart.
2. Choose Chart Properties from the shortcut menu. The Chart Properties dialog box opens.
3. Click the Chart Type icon.
   The Chart Type dialog box opens.



4. Select the type of chart we want to create.
5. Click OK.

3.3.4 Chart Properties



- **Chart Type:** Used to select different type of charts
- **Title:** Used to give a title to the chart. The values can be hardcoded as text or can come from a variable
- **Border:** Used to add a border to the chart
- **Color:** We can also specify the color associated with each data series by using the color well next to each data series
- **Legend Labels:**
  - If we choose any option other than None from the Legend list box, the Legend labels area appears on the far right of the Chart tab.
  - Legend Labels area at the default

  In the Legend labels drop-down list, there are three options:

  1. **Static** - Type in the legend text for each data series. This is the default setting.
  2. **All labels are in one array** - The legend text is retrieved from an array variable. A single variable box appears after we select this option. Specify the variable containing the legend text.
  3. **Each in different variable** - The legend text is retrieved from a different variable for each series. A variable box appears for each data series after we select this option.

3.4 FRAMES

A frame is an object that reserves an area on a page for specific message types and defines what can be placed in that area. Messages are text or graphic communications contained in a document or campaign that is placed by the Engine at run time in areas on a page held in reserve by frames.

3.4.1 Categorization of frames

3.4.1.1 Whitespace Frames

- Whitespace frames are used to fill blank areas of a design with business and marketing content. Since multiple types of information can be placed in Whitespace frames they can appear in several colors, depending on their content. Whitespace frames have a late compose time, meaning objects placed directly on the page take precedence over placement of Whitespace frames.
- Messages are dynamically placed in frames based on priority and targeting rule.
- When creating Whitespace frames to hold messages, select **Messages** from the New frame dialog box. We can customize the frame to accept only those messages we want it to hold.

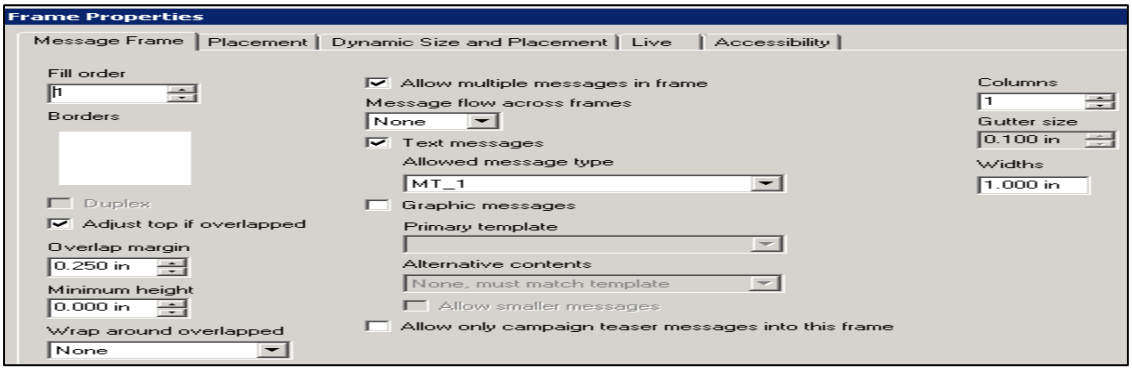
3.4.1.2 Message Frame

The Message Frame tab is the first tab of the Frame Properties dialog box. It has the same options as the Insert frame dialog box. The options on this tab enables you to select which messages are accepted, among other things.

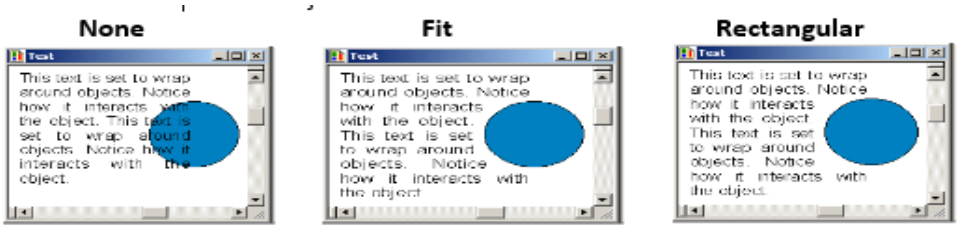
Properties of message frame

- **Message Frame tab**

1. **Fill Order:** The Fill order box is used to change the order in which frames are filled.
2. **Border:** Place lines around the frame.



3. **Overlapping:** As objects around a frame grow and move, the frame's design location may become overlapped. The Adjust top if overlapped, Overlap margin, Minimum height, and Wrap around overlapped options work together to control how the frame responds when overlapped.
4. **Wrap Around Overlapped:** The Wrap around overlapped drop-down list determines how text in the frame flows around static objects on the page. Options are:
    1. **None** - Text does not wrap.
    2. **Fit** - Text follows the shape of the other object.
    3. **Rectangular** - Text maintains a rectangular shape when wrapping around objects.



5. **Allow Multiple Messages in Frame:** Select the Allow multiple messages in frame check box if we want more than one message to be placed in the frame. If the check box is cleared, only one message is placed in the frame, even if there is room for other messages in the frame.

3.4.1.3 Content Frames

Content frames are used to embed campaign messages (graphic and text) into text boxes and tables. Content frames accept only campaign messages.

- To insert a Content frame:
  - Ensure the cursor is where you want to embed the frame. Click the button.
  - The Insert Frame dialog box opens.
  - Enter the Maximum Number of messages in frame.
  - To allow text messages, select the Text messages check box and choose an Allowed Message type from the drop-down list.
  - To allow graphic messages, select the Graphic messages check box and choose a Primary template from the drop-down list. Click OK.
  - The Embed Properties dialog box opens.
  - Select the Embed method.
  - Select other properties as necessary. Click OK.

- **Properties**
  - **Message Flow Across Frames**

  The Message flow across frames drop-down list controls the flow of messages from frame to frame and defines if objects can flow in or out of the frame. Options are:

  1. **None** - No flow in or out of the frame. The contents must fit in the frame or it does not appear.
  2. **In or Out** - Contents flow into this frame, and out into another.
  3. **In Only** - Contents flow into this frame only.
  4. **Out Only** - Contents flow out of this frame only.

3.4.1.4 Placeholder Frame

By default, the Engine starts placing imported content at coordinates 0,0 on a placeholder document page. If we want the imported content to appear at any place other than at the top left corner, use a frame with a Type of Placeholder in Designer.

Some of the characteristics of a placeholder frame include:

1. Only one placeholder frame can be placed on a page.
2. Contents imported into the frame do not split and flow to another page.

3. The frame expands vertically on the page, as needed, to accommodate the imported contents.
4. The frame must be wide enough to accommodate the text being imported.
5. These frames can have borders like other whitespace frames.
6. We can specify a rotation of the contents (at 0, 90, 180, and 270 degrees).
7. We cannot put a placeholder frame inside a table.

➢ Create a Placeholder Frame

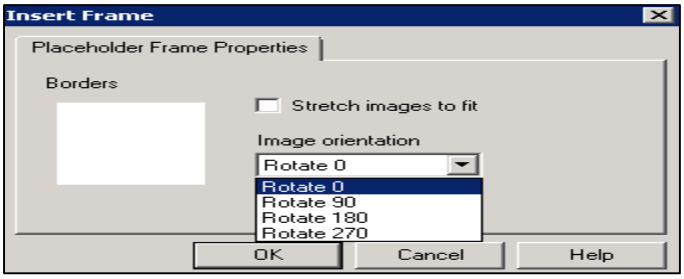
In Designer, create a new placeholder frame on a page by clicking the button and selecting the Placeholder radio button in the New Frame dialog box.

New Frame dialog box

➢ In the Insert Frame dialog box you can:

- ❖ Place Borders. Click on one or more sides of the Borders box to designate where borders should appear. Click inside the box to access the Border Properties dialog box.
- ❖ Enable an image to appear in the frame normally or select the Stretch images to fit check box to have an image fill the frame. This setting has no effect on text.
- ❖ Rotate Image Orientation by 90, 180, or 270 degrees.

Insert Frame dialog box

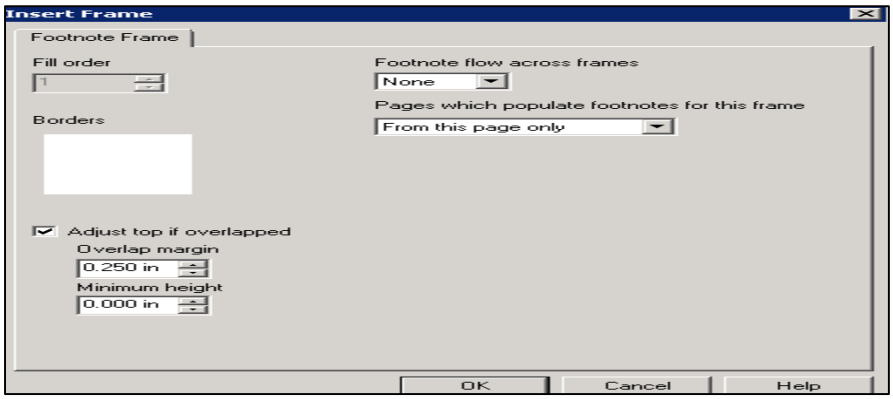


When we click OK, a purple frame displays on the page with a Document Placeholder Only label. Move the frame to the place where we want content to start on the page.

3.1.4.5 Footnote Frames

Once we have created the footnotes and have formatted them, we must decide where the text flows. Footnotes are accepted only by Footnotes frames. It is possible to create overflow pages that accept footnotes only.

Insert Frame dialog box – Footnote Frame tab



Add Footnote Frame

1. Add a Footnotes frame to the page. The Insert Frame dialog box opens.
2. To add borders to the footnotes frame, specify the borders in the Borders box.
3. If there are objects that can overlap the frame, select the Adjust top if overlapped Check box.
4. Choose how the footnotes throughout the document use this frame from the Footnotes flow across frames drop-down list.
5. Choose where the footnote content for the frame comes from using the Pages from which the footnotes can come for this frame drop-down list.
6. Click OK.
   The frame is placed on the page.
7. Drag the frame to where we want the footnotes to appear on the page.
8. Resize the frame as necessary.

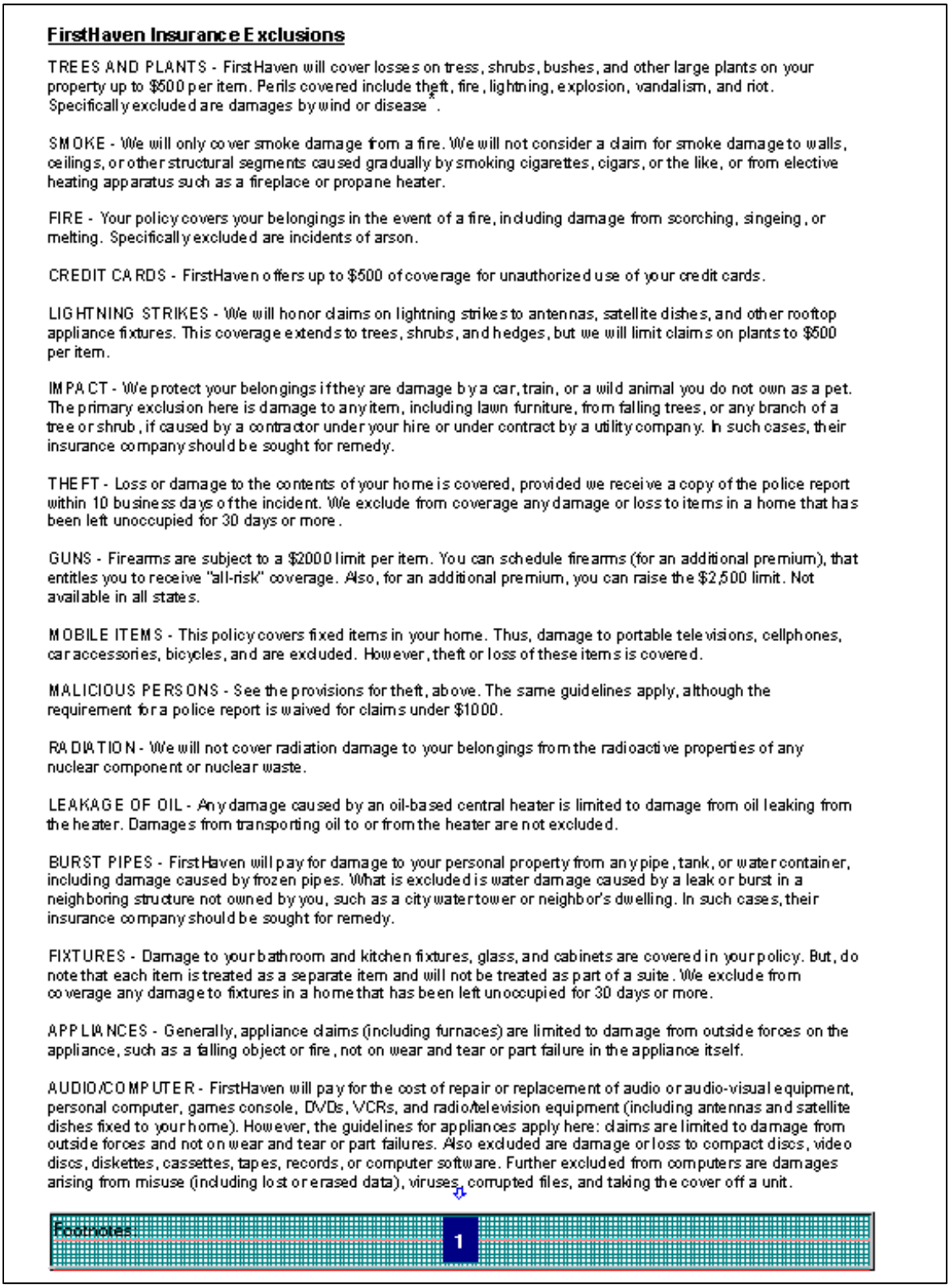
➢ Properties

- **Footnotes Allowed in Frame Area**
  The Footnotes allowed in frame area is where we control how the footnotes from the document fill the specified frame.
- **Footnote Flow Across Frames**
  Choose one of the options from the Footnote flow across frames drop-down list to control how the footnotes flow into this frame.

  1. **None** – Partial footnotes cannot flow into this frame. Footnotes placed in this frame cannot flow out to another frame.
  2. **In or out** - Partial footnotes can flow into this frame. Footnotes placed in this frame can flow out to another frame.
  3. **In only** - Partial footnotes can flow into this frame.
  4. **Out only** - Footnotes placed in this frame can flow out to another frame. Pages from which the Footnotes Can Come for This Frame Choose one of the options from the Pages from which the footnotes can come for this frame drop-down list to accept footnotes from the current page:
     - From this page only
     - From any page

- **Allow Multiple Footnotes in Frame**

  Select the Allow multiple footnotes in frame check box to allow more than one footnote in the current frame. Page with footnote frame at bottom.



FirstHaven Insurance Exclusions

TREES AND PLANTS - FirstHaven will cover losses on tress, shrubs, bushes, and other large plants on your property up to $500 per item. Perils covered include theft, fire, lightning, explosion, vandalism, and riot. Specifically excluded are damages by wind or disease*.

SMOKE - We will only cover smoke damage from a fire. We will not consider a claim for smoke damage to walls, ceilings, or other structural segments caused gradually by smoking cigarettes, cigars, or the like, or from elective heating apparatus such as a fireplace or propane heater.

FIRE - Your policy covers your belongings in the event of a fire, including damage from scorching, singeing, or melting. Specifically excluded are incidents of arson.

CREDIT CARDS - FirstHaven offers up to $500 of coverage for unauthorized use of your credit cards.

LIGHTNING STRIKES - We will honor claims on lightning strikes to antennas, satellite dishes, and other rooftop appliance fixtures. This coverage extends to trees, shrubs, and hedges, but we will limit claims on plants to $500 per item.

IMPACT - We protect your belongings if they are damage by a car, train, or a wild animal you do not own as a pet. The primary exclusion here is damage to any item, including lawn furniture, from falling trees, or any branch of a tree or shrub, if caused by a contractor under your hire or under contract by a utility company. In such cases, their insurance company should be sought for remedy.

THEFT - Loss or damage to the contents of your home is covered, provided we receive a copy of the police report within 10 business days of the incident. We exclude from coverage any damage or loss to items in a home that has been left unoccupied for 30 days or more.

GUNS - Firearms are subject to a $2000 limit per item. You can schedule firearms (for an additional premium), that entitles you to receive "all-risk" coverage. Also, for an additional premium, you can raise the $2,500 limit. Not available in all states.

MOBILE ITEMS - This policy covers fixed items in your home. Thus, damage to portable televisions, cellphones, car accessories, bicycles, and are excluded. However, theft or loss of these items is covered.

MALICIOUS PERSONS - See the provisions for theft, above. The same guidelines apply, although the requirement for a police report is waived for claims under $1000.

RADIATION - We will not cover radiation damage to your belongings from the radioactive properties of any nuclear component or nuclear waste.

LEAKAGE OF OIL - Any damage caused by an oil-based central heater is limited to damage from oil leaking from the heater. Damages from transporting oil to or from the heater are not excluded.

BURST PIPES - FirstHaven will pay for damage to your personal property from any pipe, tank, or water container, including damage caused by frozen pipes. What is excluded is water damage caused by a leak or burst in a neighboring structure not owned by you, such as a city water tower or neighbor's dwelling. In such cases, their insurance company should be sought for remedy.

FIXTURES - Damage to your bathroom and kitchen fixtures, glass, and cabinets are covered in your policy. But, do note that each item is treated as a separate item and will not be treated as part of a suite. We exclude from coverage any damage to fixtures in a home that has been left unoccupied for 30 days or more.

APPLIANCES - Generally, appliance claims (including furnaces) are limited to damage from outside forces on the appliance, such as a falling object or fire, not on wear and tear or part failure in the appliance itself.

AUDIO/COMPUTER - FirstHaven will pay for the cost of repair or replacement of audio or audio-visual equipment, personal computer, games console, DVDs, VCRs, and radio/television equipment (including antennas and satellite dishes fixed to your home). However, the guidelines for appliances apply here: claims are limited to damage from outside forces and not on wear and tear or part failures. Also excluded are damage or loss to compact discs, video discs, diskettes, cassettes, tapes, records, or computer software. Further excluded from computers are damages arising from misuse (including lost or erased data), viruses, corrupted files, and taking the cover off a unit.

Footnotes: 1

CHAPTER 4 RUNNING AN APPLICATION USING HP EXSTREAM

LEARNING OBJECTIVES: At the end of this chapter, the reader would be able to understand

- *Running of HP Exstream production Engine in windows environment, from command, in batch mode, in LINUX or UNIX environment, using shell script and also on MVS*
- *Creation and uses of control file*
- *Overview of System Key*
- *How to run the engine using control file?*
- *Details of Engine Reporting*

4.1 RUNNING THE ENGINE IN THE WINDOWS ENVIRONMENT

4.1.1 About the Production Engine

The Production HP Exstream Engine is supported in the Microsoft Windows environment. Supported operating systems include:

- Windows 2000 (Service Pack 2)
- Windows Advanced Server
- Windows Server 2003
- Windows XP

To run the Production Engine we must use Engine switches and specify them either at the command prompt when we execute the program or in a control file. The Engine reads the commands on the command line before it reads options in the control file. Thus, options we specify in the control file override the commands specified on the command line.

4.1.2 Naming Conventions

When we work from the command prompt, it is important to adhere to the naming conventions of the platform we are using.

On Windows, make the directory and file specifications of HP Exstream production file names or arguments as follows:

C:\dir1\...\dirn\filename

The drive specification, C:, is optional. If we omit the drive specification, the current working drive is assumed. Each directory specification and file name can be up to 128 characters in length.

4.1.3 Transferring Package Files

Depending on where the HP Exstream Engine is located we might need to transfer the package file to the production server. We can transfer the package file using either of the following methods:

- Use the FTP program in bin mode to transfer the file.
- Copy the file using standard Windows functionality.

4.2 RUNNING THE ENGINE FROM THE COMMAND

We can run the Engine from the command prompt using the following command:
prodengine -PACKAGEFILE=packagefilename

The mandatory -PACKAGEFILE switch tells the Engine which package file to use to compose the application. If the package file name has spaces in it, we must remember to enclose the file name in double quotes (“ “). We must specify the entire path if the package file is located in a different directory than HP Exstream.

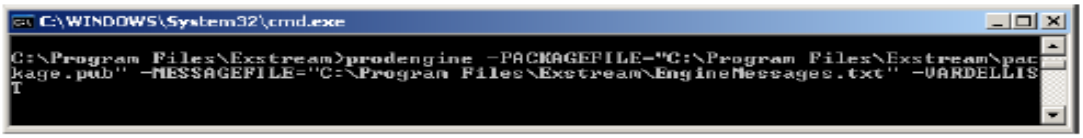
Note: *Additional switches can be used to run the Engine. We can also run the Engine using a control file. When a package file contains features the key doesn't authorize, the Engine issues a message indicating an unlicensed features and stops processing.*

4.2.1 Running the Engine

The following steps begin after we change the directory to HP Exstream's directory.
To run the Engine:

1. Type prodengine after the prompt, followed by a space.
2. After the space, add the mandatory -PACKAGEFILE switch.
   In the example the entire path is added to the specified file name:
   -PACKAGEFILE="C:\Program Files\Exstream\package.pub"
3. Enter a space after the -PACKAGEFILE switch.
4. Add any additional switches necessary to run the Engine. **Remember to enter a space between each switch.**

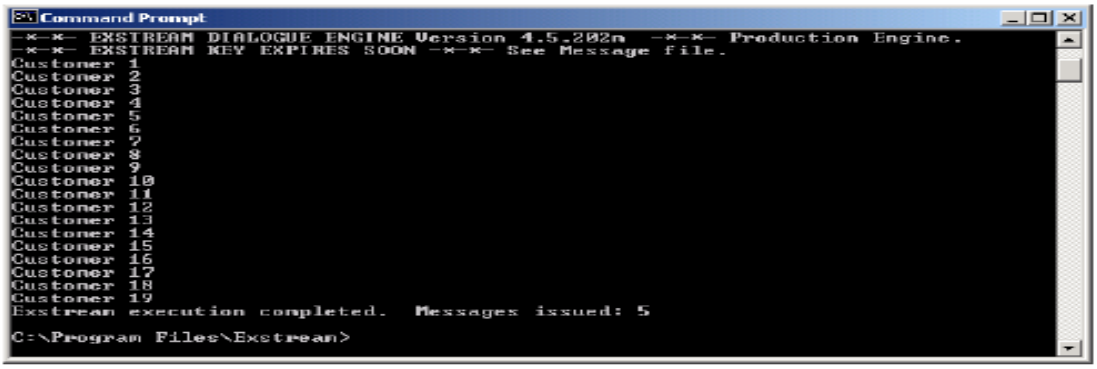
Sample Engine switches on the command line



5. Press **ENTER** to run the Engine.

The customer numbers appear in the window as they are processed.
When the Engine finishes processing, we return to the command prompt.

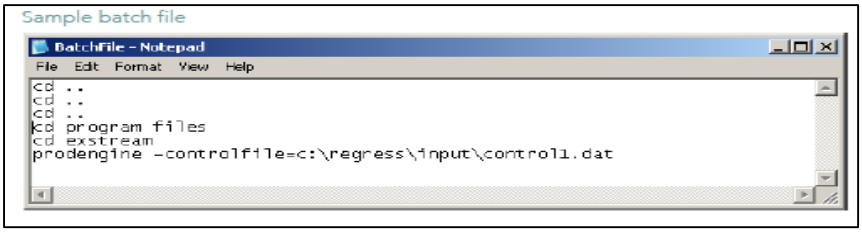
Results of Engine run in command prompt window



4.3 RUNNING THE ENGINE IN BATCH MODE

We can use a batch file to run the Engine with only a single command on the command prompt. A batch file can contain multiple commands as well as HP Exstream-specific switches to execute the Engine run.

The following is an example of a batch file. This particular batch file changes directories three times before it accesses **C:\Program Files\Exstream**. After the directory has been changed, the batch file calls the Engine to run an application using a control file.



To use a batch file, we must access the command prompt. If we have changed the directory to the batch file location, then we can type the batch file name to start the run. If the batch file is located in a different directory, then we must type the entire path of the batch file location.

4.4 RUNNING THE ENGINE IN THE UNIX OR LINUX ENVIRONMENT

4.4.1 About the Production Engine

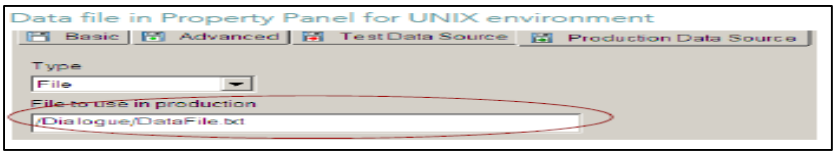
The Production HP Exstream Engine is supported in the UNIX and Linux environments. Supported operating systems include:

- IBM AIX/RS-6000 5.1 or later
- HP-UX 10.20 and above (32 bit or 64 bit)
- Sun Solaris 2.5.1
- OS/400 4.3.3
- SUSE Linux: version 8.0 (Intel), kernel version 2.4.18 for build and 2.6 for exact regression
- RedHat Linux: version 4.0 (Intel), kernel version 2.6

4.4.2 Naming Conventions

When working from the command prompt, we must retain the naming conventions for our platform. For example, we can use the following format to define the directory and file specifications of Exstream production file names: /dir1/... /dirn/filename

If we do not begin the file specification with a slash (/), the file is relative to our home directory. If we begin with a slash, the file is relative to the root directory. UNIX directory and file names are case sensitive, so verify all directories and file names for spelling and case before using them. If the names do not match, the Engine issues an error.



The figure above shows the specification for the Production Data Source on a data file for the UNIX environment.

4.4.3 Transferring Package Files

To run the Engine in the UNIX environment, we must transfer the package file to the UNIX system using our FTP program in binary (bin) mode.

4.5 RUNNING THE ENGINE ON MVS

4.5.1 About the Production Engine

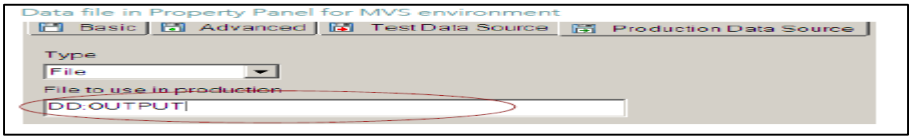
The Production Exstream Engine is supported on MVS. Supported operating systems are:

- MVS-OS/390 version 2.6
- z/OS 1.1 and above

4.5.2 Naming Conventions

It is important to remember to retain the naming conventions of the platform we are using. MVS directory and file names must be in all capital letters. For example, we use names like DD:OUTPUT or 'HLQ.XXXX.XXX' on MVS.

The production file is located on the production platform. The following figure shows the Design Manager specification for the File to use in production on the Production Data Source tab.



4.5.3 Transferring Package Files

To transfer a package file to an MVS system, upload the package file using the FTP program in bin mode.

> **Note:** *The destination PDS for the package file must have the following DCB information: RECFM=VB. LRECL=1024, Block Size=27560*
> *Failure to comply causes Exstream to read the package file incorrectly. If the LRECL setting is larger than 1024, the files are not read correctly.*

4.5.4 Using JCL Files to Run the Engine

To run the Engine on an MVS system, we must create and use a JCL file. The sample JCL file provided with the Engine is valid when we substitute the file names for the ones provided.

Sample JCL file used to run Dialogue

```
//VERAFP      JOB (999,POK), '',
//            TIME=1440,
//            NOTIFY=&SYSUID,
//            REGION=0M,
//            CLASS=A,
//            MSGCLASS=H,
//            MSGLEVEL=(1,1)
//*
//EBCEXE      EXEC PGM=ENGEXE,
//            PARM='-USECONTROL=YES'
//STEPLIB     DD DSN=P390A.EXSTREAM.LOAD ,DISP=SHR
//DLMSGRES    DD DSN=P390A.DMSGENUS,DISP=SHR
//EXMSGS      DD DSN=P390A.EXSTREAM.VERMSG(MSGAFP2) ,DISP=SHR
//BANKDATA    DD DSN=P390A.EXSTREAM.VERDATA(BANKDATA) ,DISP=SHR
//EXPACKAG    DD DSN=P390A.EXSTREAM.VERPACK(PACKAFP) ,DISP=SHR
//EXREPORT    DD DSN=P390A.EXSTREAM.VERRPT(RPTAFP2) ,DISP=SHR
//EXCONTRL    DD DSN=P390A.EXSTREAM.VMSCTRL,DISP=SHR
//EXTRACK     DD DSN=P390A.EXSTREAM.VERTRK(TRKAFP2) ,DISP=SHR
//EXOUTPUT    DD DSN=P390A.EXSTREAM.VERAFP(EXAFP2) ,DISP=SHR
```

4.5.5 Common JCL Commands

An explanation of the previous JCL example follows, beginning with line 9:

- **//EBCEXE EXEC PGM=ENGEXE**
  This command loads the Engine. The load module is the PDS, P390A.EXSTREAM.LOAD, identified by the //STEPLIB argument that follows.

- **//PARM='-USECONTROL=YES'**
  These are the parameters for the load module. In this case, the JCL tells the ENGEXE load module to use a control file. Since we did not set the name of the control file, the Engine will use the default control value of DD:EXCONTRL.
  In this example the DD:EXCONTROL references P390A.EXSTREAM.VMSCTRL.
- **//STEPLIB DD DSN=P390A.EXSTREAM.LOAD,DISP=SHR**
  This identifies the location of the load module.
- **//DLMSGRES DD DSN=P390A.DMSGENUS,DISP=SHR**
  DLMSGRES is the default DD value in MVS for the english version of the message resource file.
- **//EXMSGS DD DSN=P390A.EXSTREAM.VERMSG(MSGAFP2),DISP=SHR**
  This control value identifies the MESSAGEFILE as DD:EXMSGS. This statement equates it to the physical file MSGAFP2 in the PDS P390A.EXSTREAM.VERMSG. All Engine messages are written to this file.
- **//BANKDATA DD DSN=P390A.EXSTREAM.VERDATA(BANKDATA),DISP=SHR**
  This is the data file for the application. A value of DD:BANKDATA was entered as the name of the production file used by this application. The statement equates the logical name BANKDATA to the physical file BANKDATA in the PDS P390A.EXSTREAM.VERDATA.
- **//PACKAGE DD DSN=P390A.EXSTREAM.VERPACK(PACKAFP),DISP=SHR**
  This control value identifies the PACKAGEFILE as the DD:PACKAGE. This statement equates it to the physical file PACKAFP in the PDS P390A.EXSTREAM.VERPACK. The package file is the file that the Engine uses to create output.
- **//EXREPORT DD DSN=P390A.EXSTREAM.VERRPT(RPTAFP2),DISP=SHR**
  This control value identifies the REPORTFILE as the DD:EXREPORT. This statement equates it to the physical file RPTAFP2 in the PDS P390A.EXSTREAM.VERRPT. All Engine report information is written to this file.
- **//EXCONTROL DD DSN= P390A.EXSTREAM.VMSCTRL,DISP=SHR**
  This statement equates the logical name DD:EXCONTROL to the physical file P390A.EXSTREAM.VMSCTRL.The Engine options are read from this file when the switch -USECONTROL=YES is implemented.
- **//EXOUTPUT DD DSN=P390A.EXSTREAM.VERAFP(EXAFP2),DISP=SHR**
  This control value identifies the OUTPUTFILE as the DD:EXOUTPUT. This statement equates it to the physical file EXAFP2 in the PDS P390A.EXSTREAM.VERAFP. The Engine writes all the data destined for the output device to this file. In this example, the output is an AFP print stream.

4.6 CONTROL FILES

4.6.1 Control Files and the Command Prompt

We use the control file to send options to the Engine at run time. The control file is a text file containing Engine switches. When we use a control file, we do not have to type commands each time we run the Engine.

Any Engine switch can be placed in a control file. When we use a control file, the only command necessary at the prompt is the **-CONTROLFILE=command**.

4.6.2 Control File Types

We can use a control file for packaging or for a production run. Type the commands necessary for the production or packaging run into the control file.

4.6.3 Formatting a Control File

All switches must be:

- Preceded by a dash (-) or slash (/), depending on the platform.
- In capital letters.

4.6.4 Specifying Comments in a Control File

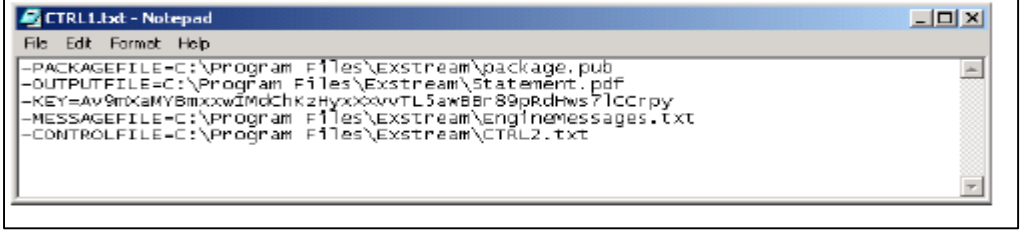
Comments are non-processing information in control files. We can use an asterisk or percent sign at the start of a line to comment out the lines of text in a control file.
For example: *comment text, %comment text

4.7 CREATING A CONTROL FILE

When we create a control file we must select the Engine switches necessary for the production run and add them to the file. To create an Engine control file:

1. Open a text editor program. In the example, Windows Notepad is used.
2. Create a file for the control file. In the example, the name CTRL1.txt was used.
3. On the first line, type the mandatory -PACKAGEFILE switch with the argument specifying the package file to be composed.
4. Specify any additional switches we want to include in the control file, listing each on its own line. Save the file.

Sample Engine control file



4.8 CONTROL FILES AND MVS

The following is an example of an MVS control file used with dynamic images.

```
-OUTPUTFILE=DD:EXOUTPUT.
-PACKAGEFILE=DD:PACKAGE
-MESSAGEFILE=DD:EXMSGS
-REPORTFILE=DD:EXREPORT
-RUNMODE=PRODUCTION
-TRACKIN=DISABLE
-TRACKOUT=FILE
-REPORT=CUSTOMER
-ENGINEDSN=P390LOC
-IMPORTDIRECTORY= EXSTREAM.TIFF
-TESTMODE
```

This control file sets the import directory using the -IMPORTDIRECTORY switch to EXSTREAM.TIFF.

> **Note:** *You must purchase the Dynamic content import module to use the* **–IMPORTDIRECTORY** *Engine switch.*

4.8.1 Specifying the MVS Name in the Control File

We can use a combination of a DD name and member name.
For example:

1. Set the DD to a PDS without specifying the member name. For example:
   //TIFFIMPO DD DSN=P390A.EXSTREAM.TIFF,DISP=SHR
2. Set the DD as the IMPORTDIRECTORY name and use the original data file.
   For example:
   -IMPORTDIRECTORY= DD:TIFFIMPO

4.9 RUNNING THE ENGINE WITH A CONTROL FILE

The following instructions utilize the -CONTROLFILE switch. The following steps assume we are working from the Exstream executable's directory. Both Windows and UNIX commands are explained below. To run the Engine using a control file:

1. At the command prompt, type one of the following commands to start the Engine run:
   - In **Windows**; the command is prodengine.
   - In **UNIX**; the command is Engine.
2. Enter a space after the Engine or prodengine command.
3. After the space, type the -CONTROLFILE command, to specify the location of the control file.
4. Press ENTER to start the Engine run.

4.10 ENGINE REPORTING

4.10.1 System Generated Files

In addition to the output, we can also generate special files when the Engine runs. We can generate message files and report files. Both types of system generated files can be used to troubleshoot problems encountered while running the Engine.

4.10.2 Message Files

Message files show us all the messages the Engine generates while processing the application.

There are four options available when we create a message file, located in the Message level drop-down list on the Run the Engine dialog box. Each option controls how much detail is provided, relating to the severity of the messages reported.

Note: *These options can also be controlled by using the* **–MESSAGEFILE** *and* **–MESSAGELEVEL** *Engine switches*

The options available on the drop-down list are:

- **All (info, warning, and error)** - All messages generated by the Engine are reported. (This corresponds with choosing I for -MESSAGELEVEL.)
- **Warning and error** - Only warning and error messages are reported in the message file. (This corresponds with choosing W for -MESSAGELEVEL.)
- **Error only** - Only error messages are reported in the message file. (This corresponds with choosing E for -MESSAGELEVEL.)

- **None** - No message file is generated.

4.10.3 Report Files

Report files show details about each object in the application.

There are four options available when we create a Report file, located in the Reporting level drop-down list on the Run the Engine dialog box. Each option controls how much detail is provided, regarding the qualification and inclusion of objects for each customer.

Note: *These options can also be controlled by using the* **–REPORTFILE** *and* **–REPORT** *Engine switches.*

The reporting level options available are:

- **None** - No report is generated. (This corresponds to choosing NONE for the –REPORT Engine switch.)
- **Selection summary** - A basic summary selection for the entire application is generated. (This corresponds to choosing SUMMARY for the -REPORT Engine switch.)
- **Customer selections** - A basic summary of the selections made for each customer is generated.
  (This corresponds to choosing CUSTOMER for the -REPORT Engine switch.)
- **Customer details** - A detailed report about all objects for each customer is generated. (This corresponds to choosing DETAIL for the -REPORT Engine switch.)

CHAPTER 5 Creating an application using HP Exstream

This document is aimed at providing an overview to create the application using Exstream. In this section we will walk through the creation of simple bank statement application in HP Exstream.

Bank Statement Application Overview

1. Bank statement is a type of report that is provided to a client that provides value and transaction information about their accounts.
2. Client may receive the statements on monthly basis, quarterly basis and also they will get a yearend statement that provides all the transaction made in that particular year.

The entire front end design for the Statements and the data displaying is taken care by Exstream. It enables to create any type of fully customized communication using data from multiple systems and information sources. Output can be prepared for delivery across virtually every print and electronic channel. Now consider we have to generate bank statement with following details.

- Portfolio Summary Page
- Detailed Summary Page for each product
- Disclosure page
- Mailer page

5.1 Create a folder

A project folder by name Bank Statement for the bank statement application is created in the Exstream Design Manager as follows:

Right click on Root Folder→New Object → Folder→ Enter Name and Description→Finish.

5.2 Developing a demo Application

5.2.1 Create an Application

An application is a container that holds all the objects needed to create a personalized communication. It must contain at least one data files and a document.

Creation and Definition of Application already discussed in chapter 2.1.

5.2.1.1 Basic Components of an Application

Following are the main components of an application:

1. Data Files

We must have at least one data file in our application.

The Sample columnar data file format is as shown below.

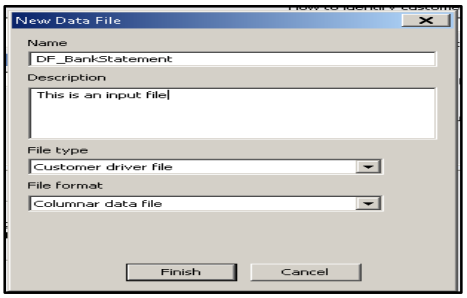
```
1A
2A20150115201501162015011 9
3A00100089999999999900002014112620150116001000000000000999999999THOMAS B MAX
4J 00100000000000999999999
4E 00100000000000999999999MAX                          THOMAS                B
5YNNN  NN  NN
6CNNNet Amount Invested Since                            20150118......
5BTOTAL PORTFOLIO                          N     ..................................
6ETHOMAS B MAX
7DN     NNNCERTIFICATE-1 YEAR                                                0010000000
7DN     NNNCERTIFICATE-2 YEAR                                                0010000000
4C
5E20141126201501 6Certificate-1 Year
6J00020141128Initial investment                                  ..................
6H20141126000000000.....................................................002100002
5E20141226201501 6Certificate-2 Year
6J00020141229Initial investment                                  ..................
6H20141226000000000.....................................................002100002
```

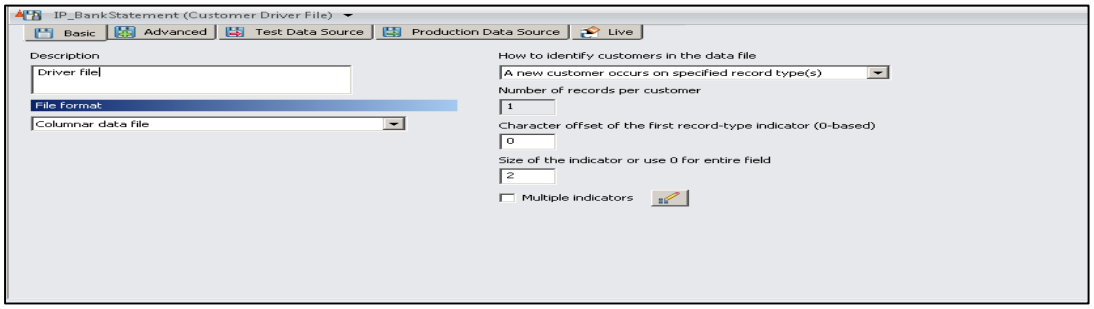
Below is the Input data record structure.

- 3A record type indicates the start of the new customer. Under 3A, the 4J, 4E, 5Y, 7D records define data for portfolio summary page.
- Every product is identified with the specific record type and new section will be created for each product.

Customer Data file

We have created a columnar customer data file and properties are set as required.





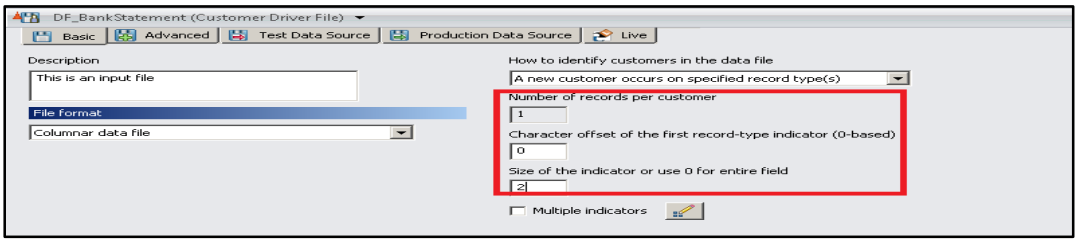
- **Basic Tab:**

How to Identify Customers in the Data File:

In this drop-down list, for our demo, the option" **A new customer occurs on specified record type(s)**" is selected.

When this option is selected, following fields are specified:

- Beginning location of the first record type indicator in the Character offset of the first record type indicator (0-based) box. **For our demo, it is given as 0.**
- Length of the record type indicator in the Size of the indicator box. **Here, the length is given as 2.**

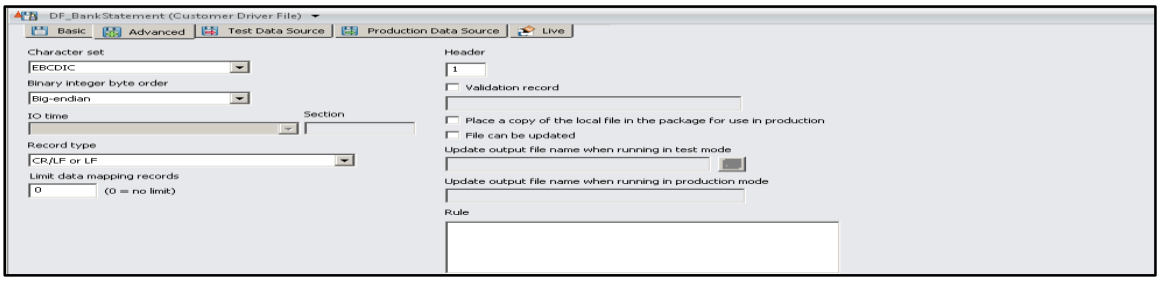


- **Advanced Tab:**

For this demo following Options are set for a data file object on the advanced tab.

- **Character Set:** Here the option "**EBCDIC**" is selected**.** This option is selected if the data Source contains mainframe-specific formats such as packed decimal or zoned.

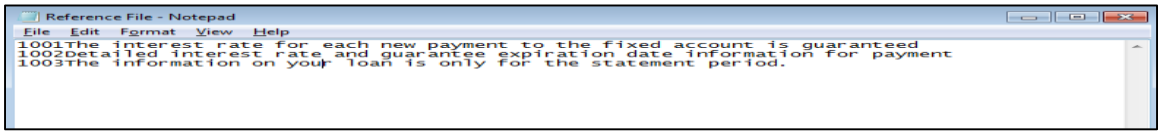
- **Binary Integer Byte Order:** Here, the option "**Big-endian**" is selected, which means" Exstream converts the file to big-endian order".
- **Record Type:** Here, the option **"CR/LF or LF"** is selected, which means "Either a carriage return and line feed, or just a line feed, separates one record from the next."
- **Header***:* Number of records the headers occupy in the File is entered in the Header box. **Here, the value of the header is set to 1.**



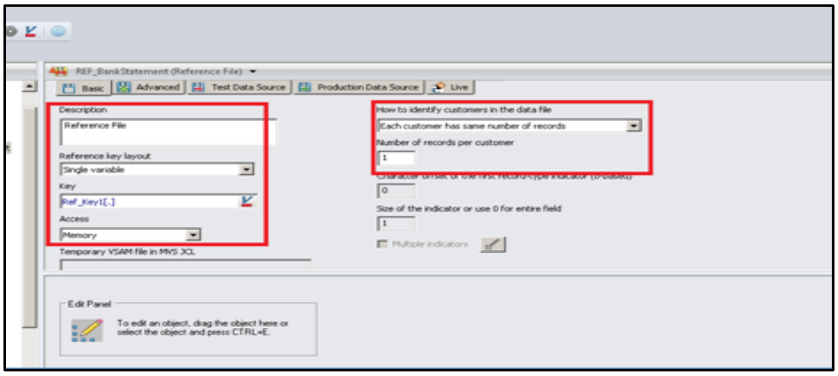
- **Test Data Source Tab:**
  - **Type:**
    Here, the option "File" is selected, which means, the engine receives data from an external file. This is a default setting.
  - **File to Use for Data Mapping:**
    In the File to use for data mapping box, the external file used to provide information to the data file for testing purposes is specified.
- **Production Data Source Tab:**
  - **Type:** <same as above Test Data Source Tab >
  - **File to Use in Production:** The production DD name taken as DD:DRIVER.

Reference File:

In our application, we use reference file for displaying messages based on the message code. The structure of the reference file is as shown below.



Here the four digits are mapped as the reference indicator. And reference key will be message Id which will be passed in the driver file. This will trigger the corresponding message. **<Reference file creation already discussed in chapter 2.2.>**



1. **Basic Tab:**
   - **Reference Key Layout:** Here, the layout is given as “Single Variable”.
   - **Key*:*** Here, the variable Ref_Key1 is selected as the reference key variable.
   - **Access:** Here, the option “ Memory” is selected.
2. **Advanced Tab:**
   Here, in the demo, the following properties have been set:
   - **Character Set:** The character set is set as EBCDIC.
   - **Binary Integer Byte Order:** The binary integer byte order is set as Big-endian.
   - **Record Type:** The record type is selected as “CR/LF or LF”
   - **Limit Data Mapping Records:** The number of data mapping records is limited to 1000.
   - **Header:** Header is set at 0.
   - **Rule Run Time:** Rule Run Time is set at “Once at Startup”
3. **Production Data Source Tab:** Here the file provided is: **DD:REFMSG.**

NOTE: In this demo, we will not be covering details of initialization file and report file.

2. DOCUMENTS:

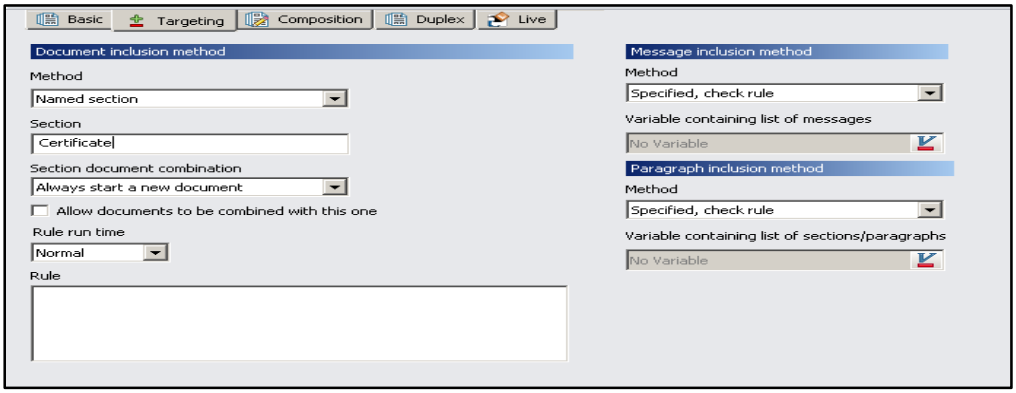
For every application, there should be a minimum of one document available to the application. Once the pages are created, they can be dragged in to the document as shown below:

For this application, we have created the four documents as shown below:

Each document will be triggered based on the named section in the targeting tab.

<Creation of document has been discussed in chapter 2.1>

In demo, the method selected is “**Named section**”.



3. PAGES:

A document must have at least one page to be a valid document. We can also drag messages and sections to the document to add them.

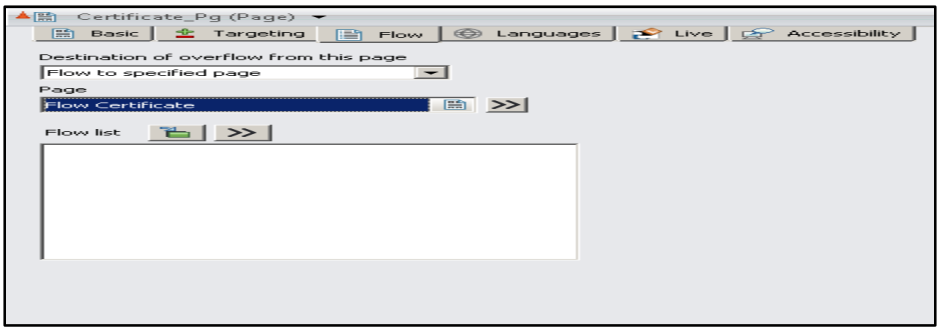
<Creation of pages has been discussed in chapter 2.1>

Following are some of the properties that need to be set for a page:

- **Paper type**: Can be simplex, duplex, etc.,
- **Orientation**: Can be Portrait, Landscape, Portrait reversed & Landscape reversed. The "**landscape**" orientation is selected here, which means the longer side of the page runs from left to right and the objects on the page print as ordered.

The **targeting tab** can be used to insert rules for the page, if any.

The **flow tab** can be used to determine if there is any overflow required or not. Here the option "Flow to specified page" is selected, which means, Overflow is placed on another page. Select the page from the Page box.



Flow page :

Exstream enables us to automatically control the number of pages needed for content. There are two main terms we should know in order to understand Flows:

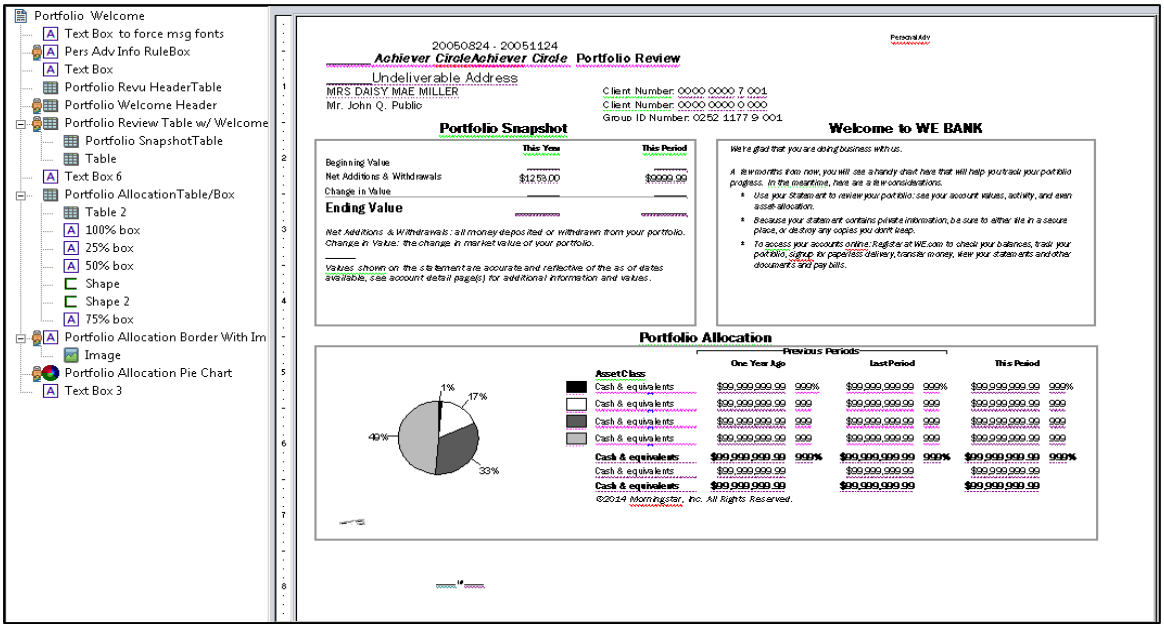
a. **Flow Page:** is a page that accepts overflow content from another page in an application.
b. **Frame**: is an object that reserves an area on a page for specific message types and defines what can be placed in that area.

Below is the flow page design for certificate page

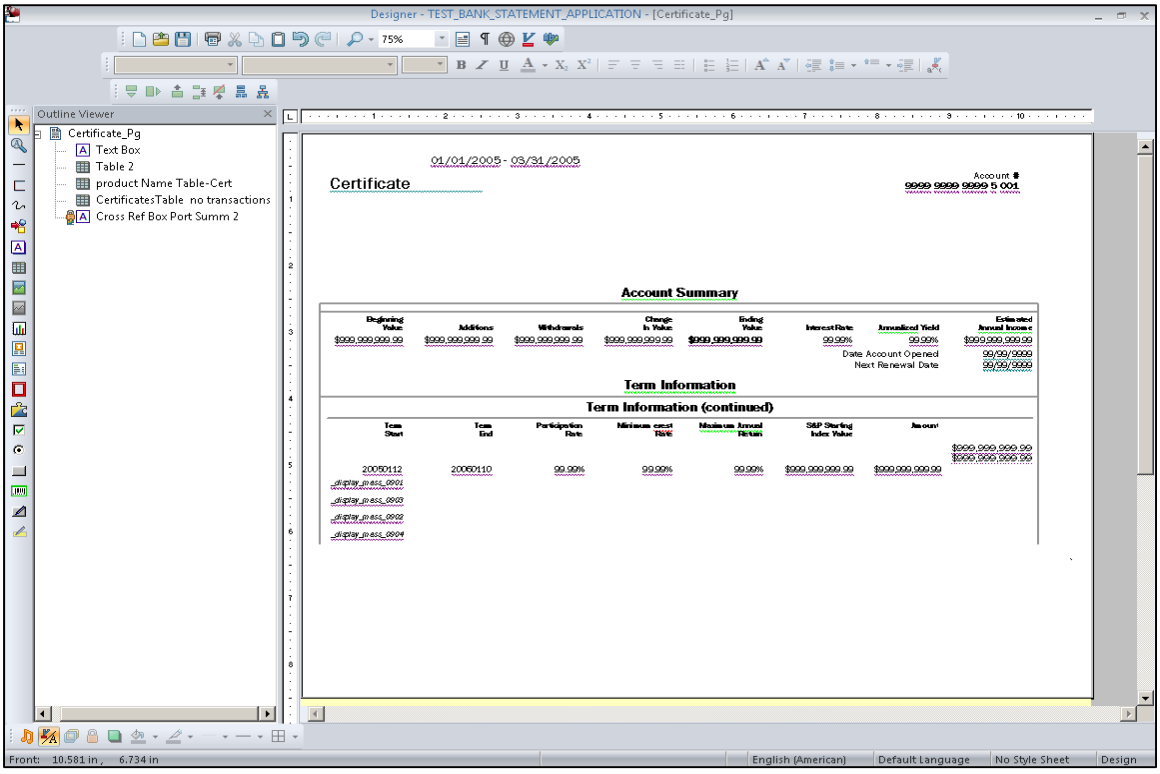
To edit the contents in a page, drag the page to the Edit Panel. This will initiate the Exstream Designer.

Using the Exstream designer the following pages are created for this Bank Statement application.

Portfolio page:



Certificate detailed Page design:



4. Output Queue

For every output file being created from an application, there needs to be queue defined to route the output files from the Exstream to the physical location.

<Creating Output and Output Queue already discussed in chapter 2.3>

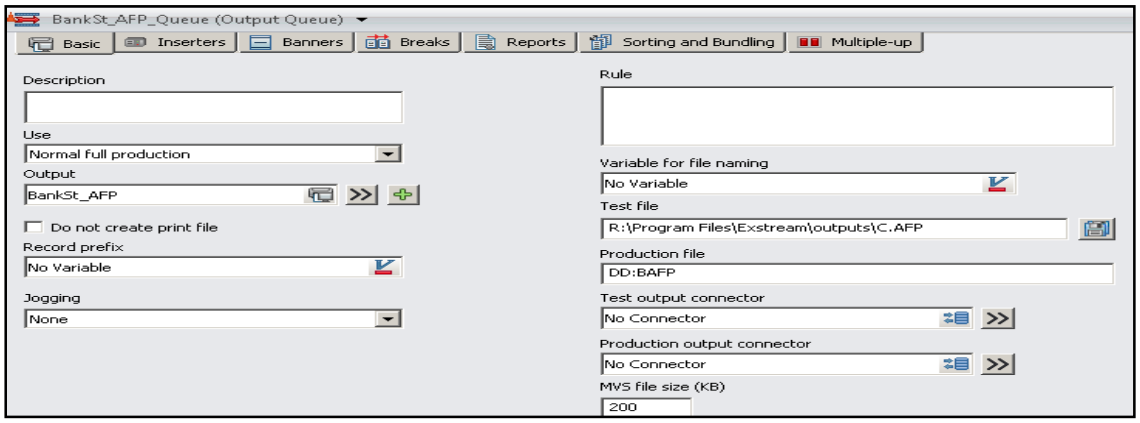
An output queue needs to be added to get an output- **BankSt_AFP.**

Output queue is then added to the application based on the output required:

BankSt_AFP_Queue

In the basic tab,

- output is selected as “BankSt_AFP”
- In the test file, location for the AFP output is provided
- In the production file the logical name is given as : DD:BAFP
- In the rule tab, required rules are provided



5. **Packaging**

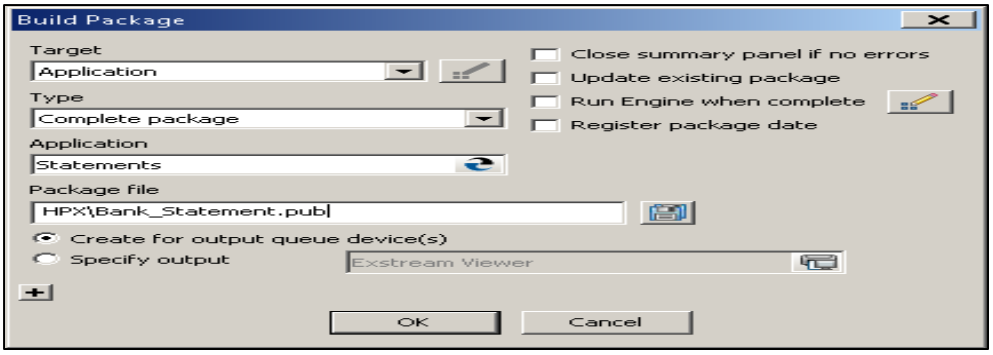
   When we package an application, the result is a package file (sometimes called a pub file) that combines objects such as: Data files, Output Devices or Queues, Application-specific variables, Documents, Pages, Sections and Paragraphs, Campaigns and campaign messages.

Build a Package File:

Package files are created and modified exclusively in Design Manager.

When we create a package file, we:

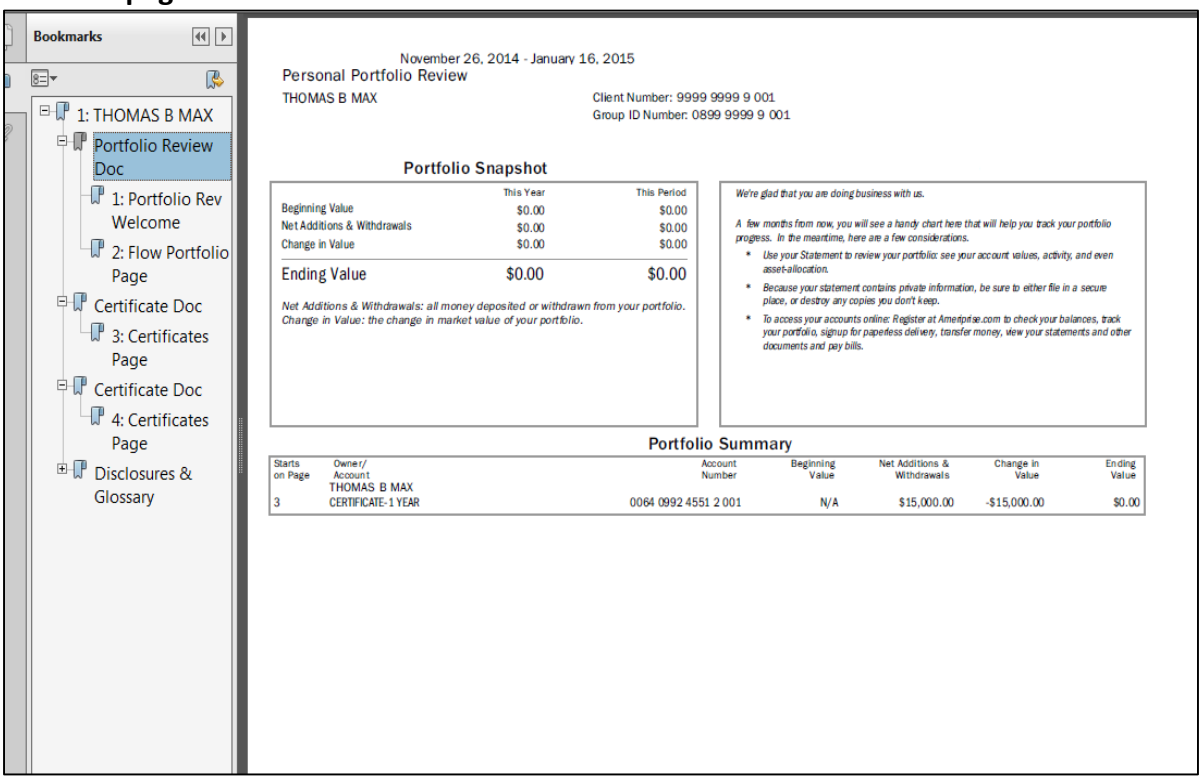
1. Select the **application** to package.
2. Right click the application to package and select **Package Application**. The **Build Package** dialog box opens. From here we can select various packaging options.



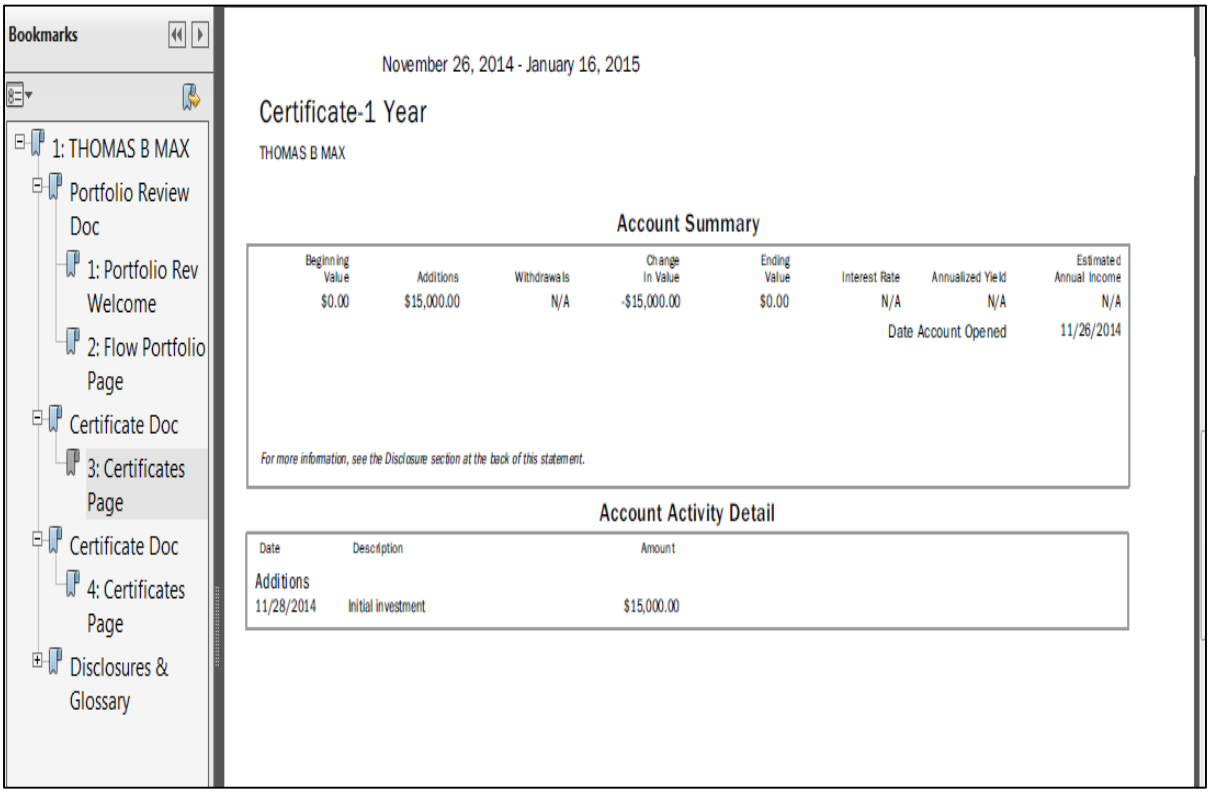
Click **OK**. The package file is created and saved.

Below is the screenshot of the output created using the package file.

Portfolio page:



Certificate page:



REFERENCES

- HP Exstream user manuals.
- http://www8.hp.com/us/en/software/exstream-software.html?compURI=1384727