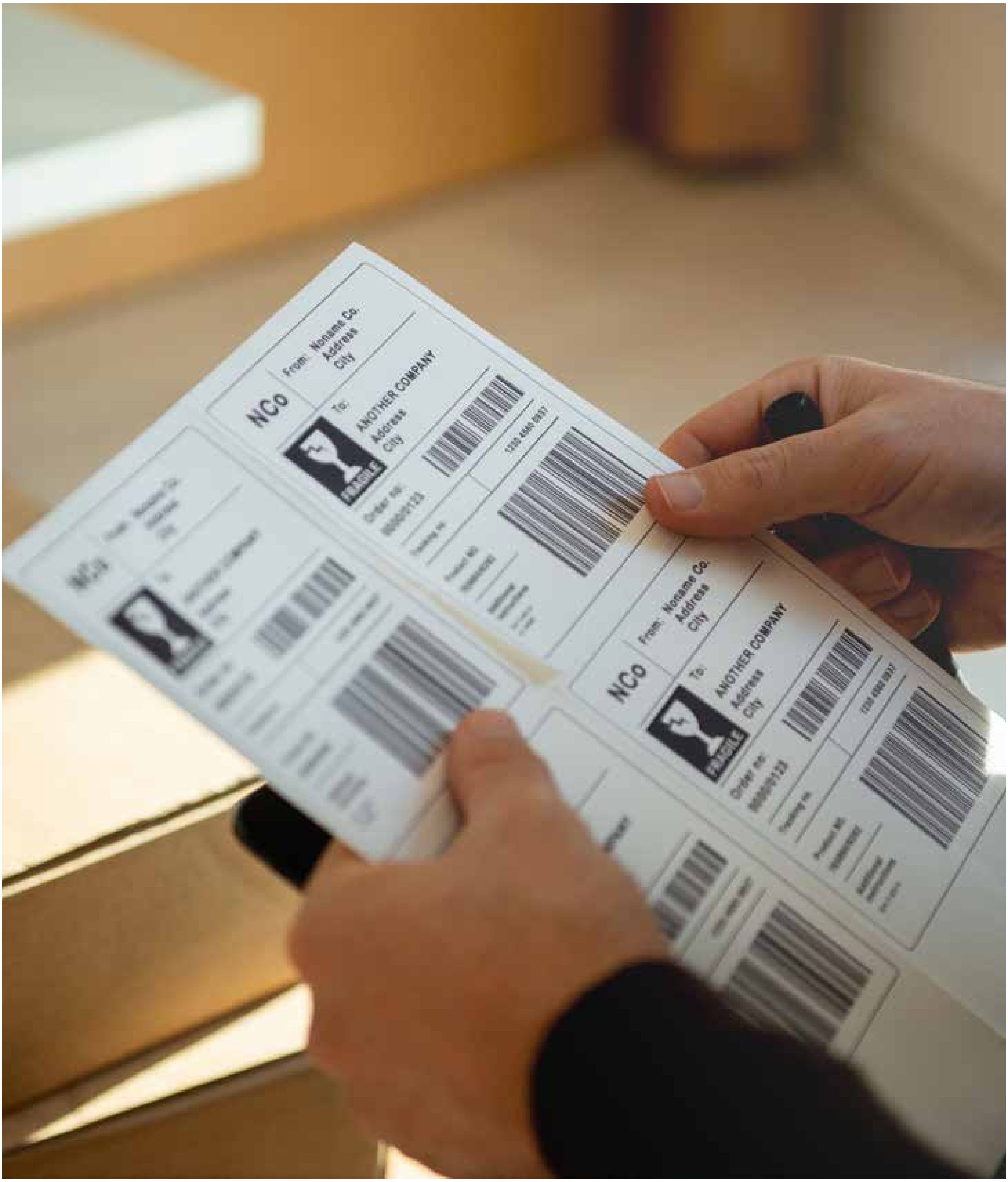
Label Artwork Template Generation Using Cognitive Automation

Whitepaper by Narayan Mishra, Assistant Manager – Data Science | Faustina Selvadeepa, Senior Associate at NEXT Labs



Contents

1.
Business Context

Labels are an integral part of any product delivery. Labels provide the branding, promotion, information on usage, safety, contact, other compliance and regulatory content. Packaging products in many industries like pharmaceuticals, healthcare devices, chemicals, food, etc., are heavily regulated. Any issue with the labels can bring about compliance, financial and reputation risks for the manufacturers making managing labels a critical task. These labels can have a wide variety ranging from small and simple black-on-white background labels to packaging labels with rich and complex content to multipage instructions for use.

Label management involves creating, managing and printing labels. There are organizations that specialize in managing labels for their end clients. They adopt specialized processes and proprietary systems to manage and print labels as per their customers' requirements.

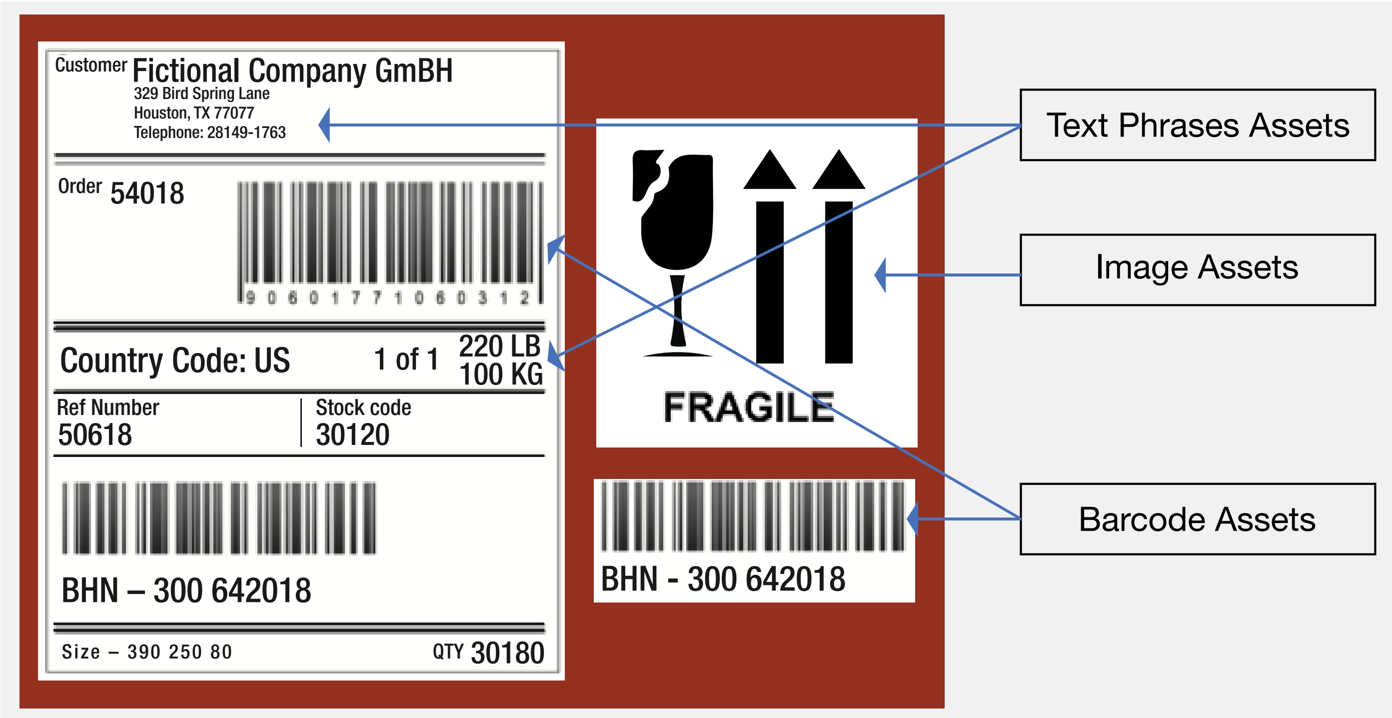
2.
The Business Challenge: Need for Cognitive Automation

Label migration is an important process of label management. This usually happens when the label management company is tasked with managing the existing labels of the client organization. In the process, the existing label designs are acquired and onboarded on the proprietary label management system. This involves identifying the assets on each of the labels (text phrases, images, barcodes, etc.), creating an inventory of these assets and templatizing the common label formats. Due to the proprietary nature of the label management systems, the digitized templates need to be in their specific formats. Usually, the templates are created using Extensible Markup Language (XML), which has information about the type and location of the various assets on the label and their formatting information. Template generation has been traditionally done manually, with a specialist taking out all the information from the existing label images and codifying them in XML on a platform. This makes it a very effort-intensive, operationally costly and difficult to scale process. A skilled specialist can take anywhere between a few hours to a couple of days to create a template depending on the complexity of the label.

An automation solution will help in reducing the effort and the cost of operations as well as reducing the lead time for template generation, thus providing faster time to market. Given the unstructured nature of the labels as well as the high variety, any rule/logic-based automation will not be applicable. The automation requires cognitive components which can make sense of the unstructured documents and give the information in them a standardized structure that can be translated to the required XML code.

3.

The Solution Approach



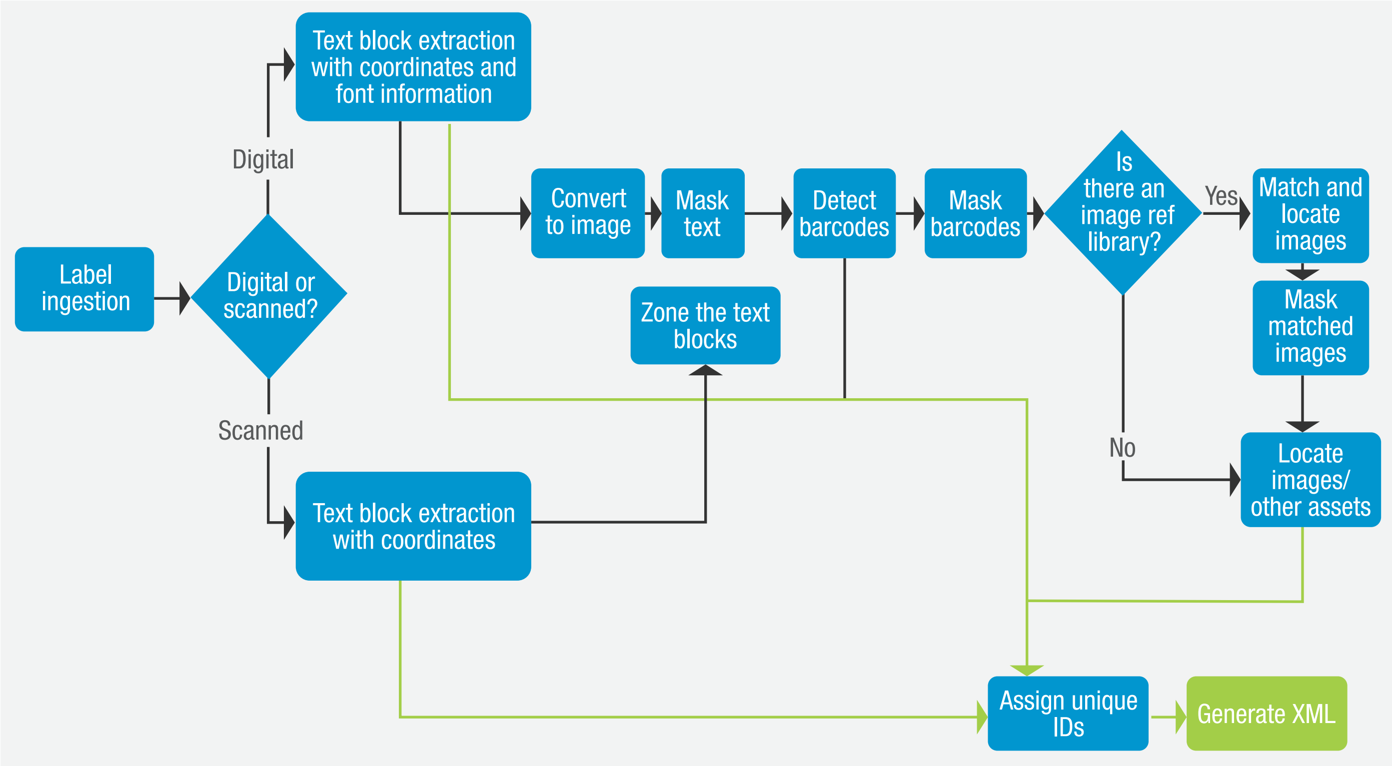
Mphasis has configured the DeepInsights™ platform to identify the assets in the labels and create the assets inventory. The platform is of modular nature, with modules performing specific functionalities and orchestrated by a workflow to achieve the task. Along with the asset inventory creation, the modules also bring in additional capabilities which gather other metadata about the assets. The "Template Generator" was designed to use these metadata to collect the information required to generate the templates.

To develop an XML template for a single label, a data frame (called Assets Sheet) was created that had all the necessary asset information. This Asset Sheet has one row and a column for each asset needed for XML. Assets are classified into three high-level categories: text, image and barcode. There are further two types of assets in the Text category: Special Text (text in larger than normal font size) and Content (text in normal font size). All four border box points encircling the asset, font information (for text category), and asset ID are required (unique for each asset within a single label).

The Workflow

Three distinct approaches were adopted, one for each asset type, and created specific containerized cognitive automation modules for each asset category. The container automation modules were deployed as a service on the workflow, called in a loop, according to asset categorization. A final script combines all the information to generate the root XML element concatenated with all the module XML containers.

A label has multiple assets, and these assets can be classified into the following categories: Special Text, Content, Image and Barcode. The idea is to capture all the assets that are available in a label and tag it with unique IDs under each category. In addition to capturing the location of the assets, we also snip out the assets and store them in a repository for future reference.



4.

Capturing the Assets and their Attributes

Specific strategies/approaches were adopted for identification, collecting metadata and classification of the Text, Image and Barcode Assets.

Text Content

The Template Generator ingests labels in the form of digital PDF and scanned PDF files, and we employed two separate methodologies for these two file types. We used python based opensource packages to extract text from digital PDF files. The scanned PDF files are converted into PNG files and processed using Google Vision API to extract text from images, along with paragraph and line attributes. The letters were merged to form words, and words to form lines. Using bounding box coordinates, the text height for each line was estimated and lines with comparable text heights inside were grouped to recreate paragraphs. These groupings were termed text zones and categorized into known and unknown fields. Likewise, text zones in Digital PDF files were also categorized.

Barcodes

To better understand barcode identification processors, it can be divided into two major categories: Training and Prediction. Data for 1-D barcodes, 2-D barcodes and images with no barcodes were collected for training. Following data collection, a multi-label image classification model was trained using PyTorch. It uses a combination of OpenCV, custom-trained PyTorch-based Deep Learning models and an open source barcode decoder for inference. For each batch of labels, first boundary boxes around assets were generated in a label image using OpenCV-based dilation and erosion algorithms with a custom kernel. Further, the images were processed using OpenCV.

Next, the snipped images were filtered within the boundary boxes based on the area threshold and passed to the image classification model for classifying into a 1-D barcode, 2-D barcode or no barcode. The barcode decoder was used to avoid false positives as the predicted barcodes were passed through it. The decoder also decodes barcode information and barcode type (EAN, data-matrix, etc.)

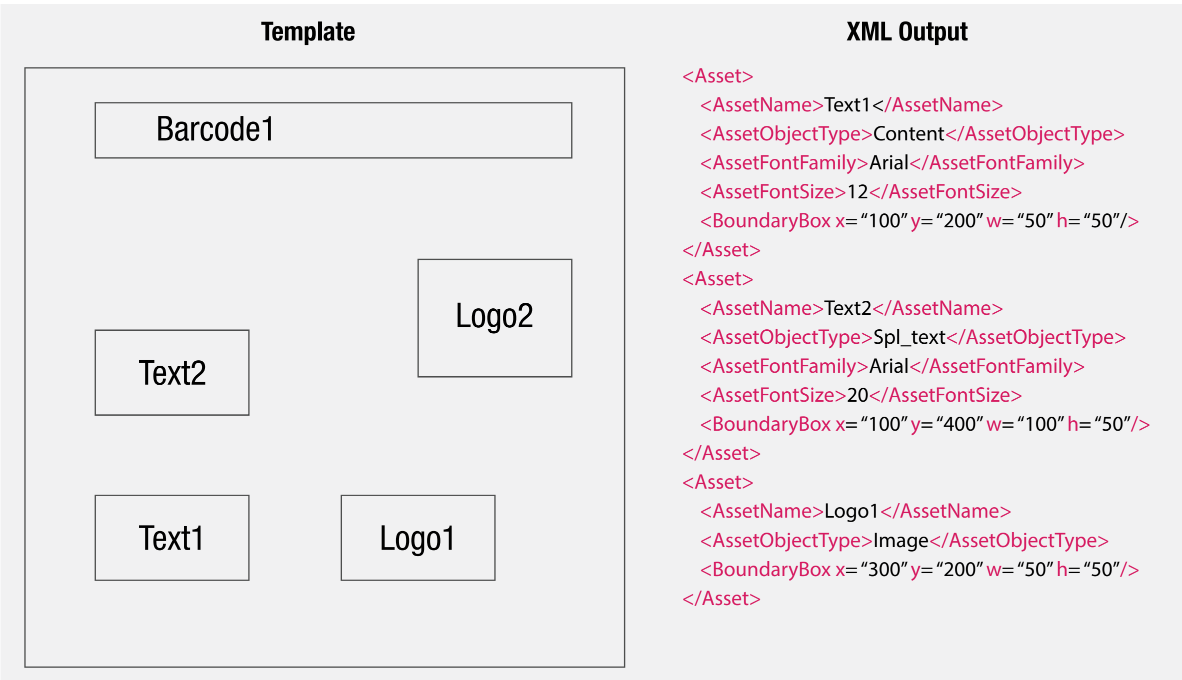
Images

There are a series of steps that takes place to identify images in a label. First, the non-image attributes in a label is identified, the respective coordinates are obtained and then masked using the background color of the label. This way, the dieline, text and barcodes are masked leaving behind only image assets. We pass this label having only the images into the image identification modules. Two approaches are used for image identification. First, the labels pass through an image matching module. The image matching module is based on feature point detection and description algorithm with scale invariance and rotation invariance. This requires an image reference library. If the image reference library is not available, or not all images on the label are matched, then an image detection module is executed. The image detection module uses a KNN classifier and PyTorch-based models to detect and classify the image assets.

5.

Generating the XML Template

A python script gathers the data from the modules explained before and inserts them into the XML tree to generate the XML-based template code. The XML template contains information about all the attributes in a document including the font type, style, etc.



To generate the XML template, the data frame “Assets Sheet” provides all the metadata, i.e., asset IDs for all the assets (barcodes, images, special text and content), their boundary box coordinates and other formatting information such as font size, orientation, etc., captured from various modules to the script. Each asset has a custom tag associated with it to identify it as a particular type of asset. The script then places them in their respective containers before appending to root XML element. The output of the script is the Label Template in the XML format which can now be imported into the Label Management Platform.

6.
Way Forward

The DeepInsights™ Label Template generator is one of its kind cognitive automation solution for a traditionally high-skill, high-effort task. This proof of concept demonstrates the possibilities of using advanced cognition, Machine Learning, natural language processing and image analytics to convert a highly unstructured data like a label to a very structured code. With the established technology and solution approach, Mphasis is working on the following steps.

1. Adding enhancements to the solution for processing more complex labels and more granular classification of the assets.
2. Capturing more metadata from the label assets like font sizes, color information, etc.
3. Converting the solution to an API-based web service that can be integrated with the label migration/management platform.
4. SaaSification of the solution allows it to be available for consumption by the label management companies while Mphasis maintains and supports the solution.

Authors

Narayan Mishra

Assistant Manager – Data Science

Narayan Mishra is an Assistant Manager – Data Science at Mphasis with 3+ years of professional experience. He completed his Master of Technology from IIT Kanpur in Industrial & Management Engineering. He is proficient in Operations Research and Statistical Modelling. He is interested in various Machine Learning & Deep Learning areas like product recommendations, natural language processing, and computer vision. He has also been actively involved in building a production scale recommendation system for a large organization. He is exploring workflow management platforms for data engineering pipelines to programmatically author, schedule, and monitor workflows. He has a keen interest in Quantum Computing and Privacy Preserving Machine Learning. .

Faustina Selvadeepa

Senior Associate at NEXT Labs

Faustina Selvadeepa works as a Senior Associate at NEXT Labs. She has around 10 years of industry experience and holds a master's degree in computer application. Her research areas include Natural Language Processing and Machine Learning. She has worked on various client PoCs and helped in building the Mphasis IPs, PPML, PACEML and InfraGraf®.

About Mphasis

Mphasis' purpose is to be the *"Driver in the Driverless Car"* for Global Enterprises by applying next-generation design, architecture and engineering services, to deliver scalable and sustainable software and technology solutions. Customer centricity is foundational to Mphasis, and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized (C = X2C$^{2}_{TM}$= 1) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization, combined with an integrated sustainability and purpose-led approach across its operations and solutions are key to building strong relationships with marquee clients. Click here to know more. (BSE: 526299; NSE: MPHASIS)

For more information, contact: marketinginfo.m@mphasis.com

USA
Mphasis Corporation
41 Madison Avenue
35th Floor, New York
New York 10010, USA
Tel: +1 (212) 686 6655

UK
Mphasis UK Limited
1 Ropemaker Street, London
EC2Y 9HT, United Kingdom
T : +44 020 7153 1327

INDIA
Mphasis Limited
Bagmane World Technology Center
Marathahalli Ring Road
Doddanakundhi Village, Mahadevapura
Bangalore 560 048, India
Tel.: +91 80 3352 5000

NR 06/10/22 US LETTER BASIL7584

www.mphasis.com