AFRL-RH-WP-TR-2021-0069

Strategic Policy Analysis Resource and Knowledgebase (SPARK) Version 2.0 System Design and Code Production

Johnny J. Weissmuller
Air Force Personnel Center
Strategic Research and Assessment Branch

Kristin A. Cazares
Osi Vision, LLC

September 2021
Interim Report

AIR FORCE RESEARCH LABORATORY
711TH HUMAN PERFORMANCE WING,
AIRMAN SYSTEMS DIRECTORATE,
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RH-WP-TR-2021-0069 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signature//

THOMAS R. CARRETTA, PhD
Work Unit Manager
Performance Optimization Branch
Airman Biosciences Division

//signature//

R. ANDY MCKINLEY, DR-III, PhD
Core Research Area Lead
Cognitive and Physical Performance
Performance Optimization Branch
Airman Biosciences Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
15-09-21	Interim	August 2020 to August 2021

4. TITLE AND SUBTITLE
Strategic Policy Analysis Resource and Knowledgebase (SPARK) Version 2.0 System Design and Code Production

5a. CONTRACT NUMBER
FA8650-20-F-6233

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62202F

6. AUTHOR(S)
Johnny J. Weissmuller* & Kristin A. Cazares**

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
_H12D

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
**Osi Vision, LLC
100 Sandau Rd., Ste.210 SanA,
San Antonio, TX 78216

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command
Air Force Research Laboratory
711th Human Performance Wing
Airman Systems Directorate
Airman Biosciences Division
Performance Optimization Branch
Wright-Patterson AFB, OH 45433

*Air Force Personnel Center
Strategic Research and Assessment Branch
AFPC/DSYX
550 C. St. W, Ste. 45
Randolph AFB, TX 78150

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
711 HPW/RHBC

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RH-WP-TR-2021-0069

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A: Approved for public release.

13. SUPPLEMENTARY NOTES
Report contains color. AFRL-2021-4120 , cleared 17 November 2021

14. ABSTRACT
This report documents the design effort and code production for the current Strategic Policy Analysis Resource and Knowledgebase (SPARK) Version 2.0 system, authorized by Air Force Manual 36-2664, Personnel Assessment Program, dated 16 May 2019. The purpose of SPARK is to support strategic studies by the Air Force Examining Activities Program (PE 81713F) to evaluate current or proposed personnel policies to improve "person-job-match" and general human capital management from pre-accession through retirement.

15. SUBJECT TERMS
SPARK, DRILS, DICE, FIDO

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Thomas R. Carretta
Unclassified	Unclassified	Unclassified	SAR	65	19b. TELEPHONE NUMBER *(Include Area Code)* N/A

Table of Contents

PREFACE

Per Air Force Manual (AFMAN) 36-2664, Personnel Assessment Program, dated 16 May 2019, the Air Force Personnel Center/Strategic Research and Assessment branch (AFPC/DSYX) is tasked with supporting the effort to optimize the United States Air Force (USAF) mission readiness and to facilitate the total force capability in recruiting, selection, classification, training, promotion, utilization, and retention. AFPC/DSYX provides both broad institutional support and targeted process improvement to the USAF. The Strategic Policy Analysis Resource and Knowledgebase (SPARK) team within AFPC/DSYX maintains the SPARK system that supports strategic policy analyses by the USAF Examining Activities program, located in Air Force Personnel Research, Testing and Survey Policy (HQ AF/A1PT).

A primary mission of the SPARK is to support the ongoing measurement and predictive analyses that ascertain the optimal fit of people, given particular attributes, into both the Air Force in general (accessions) and into specific jobs or career fields (classification). This umbrella mission has been pursued for the past 73 years in the Air Force personnel research community and is colloquially called "person-job-match." The AFPC/DSYX SPARK team and industrial/organizational (I/O) psychologists perform the analyses to evaluate Air Force personnel policies in order to shape future human capital management operations. This effort is policy analysis "due diligence", as envisioned in the Dixon Plan in September 1978 (USAFPP- 1, The USAF Personnel Plan – Volume One – Personnel Management Objectives, 29 September 1978): Concept 7: "*Objectives, policies, practices, and procedures must reflect integrity, be authoritative, and be fully supported by sound rationale and a research and analysis capability."*

The structure of the SPARK system is driven, in part, by the demands of these evidence-based, data-driven studies performed by the I/O psychologists in HQ AFPC/DSYX. The SPARK system has three major components: the huge resource of a 50-year longitudinal databank of databases, state-of-the-art analytic methodologies supported in software, and a knowledgebase of scientific reports, historical studies, and analytical methodologies.

The cornerstone of the SPARK Databank is built with hundreds of Master and Project data files (800+ CD-ROMs) from more than three decades of personnel research at the Air Force Human Resources Laboratory (AFHRL). The 30-year legacy (1968-1998) of AFHRL's Human Resources Research Databank (HRRD) still drives many of the current SPARK Databank procedures (i.e., file structure and layouts, data cleaning procedures, cosigned Data Use Agreements (DUAs), data-pulls and update procedures). Details of these inherited and modified processes can be found in the SPARK 1.0 Report (28 JUL 2020) as exemplars of the historical and current operational SPARK data processing practices. The existing functional SPARK 1.0 system and procedural practices annotated in the aforementioned report are the basis for the direction and recommendations for SPARK II hosting and programmatic efforts detailed in this SPARK 2.0 document.

1.0 SUMMARY

This SPARK II report uses the documentation and data management practices of the SPARK Version 1.0 System authorized by AFMAN 36-2664, Personnel Assessment Program, dated 16 May 2019, as its basis for the recommended integration of the current SAS 9.4 software system with open-source solutions and the anticipated SAS Viya platform. As background, the SPARK databank supports AFPC/DYSX's strategic studies to evaluate the effectiveness of personnel measurement, selection, and classification methods from pre-accession through retirement. Results of these studies provide support to the Strategic Personnel Assessment Program (HQ AF/A1PT) in development of Air Force policy. The SPARK databank is the successor to the AFHRL HRRD, which spanned from 1968 to 1998. SPARK procedures are driven in large part by the legacy HRRD system. Details of these inherited and modified processes are provided by Weissmuller and Cazares (2020) as exemplars of the historical and current operational SPARK data processing practices. The focus of the SPARK II project began as a design effort, an investigation of emerging technologies for SPARK II hosting with a recommendation for the way forward for code production that was anticipated for Phase III of the SPARK project. However, Information Assurance (IA) requirements shifted the project timeline left and the Phase II design effort also included the code production that enabled the legacy SPARK 1.0 system to run natively in a Windows 10 environment, which no longer required the use of offline Windows XP machines or XP-emulators. Consequently, at the conclusion of SPARK Phase II, the updated SPARK 2.0 system is capable of running on the Air Force network. Furthermore, at the time of this writing, the developmental sandbox for the SAS Viya platform for the AFPC and their partners (A9, the Air Education & Training Command/Studies & Analysis Squadron (AETC/SAS), and the Air Force Reserve Command (AFRC)) is anticipated to arrive in September 2021.

2.0 INTRODUCTION

The mission of the Air Force Personnel Center's Strategic Research and Assessment Branch is to improve the compatibility between people and Air Force career fields (i.e., person-job-match) to improve human capital management for optimal Air Force mission capability. Phase I of the project was a development effort to record the existing, functional SPARK 1.0 procedures. Phase II uses the Phase I documentation as a springboard to envision a 21st century design for the SPARK 2.0 System that manages (imports, protects, stores, retrieves, and analyzes) the Air Force's historical datasets and also integrates newly emerging datasets from tests and assessments developed by the AFPC/DSYX, AFRL, their contractors, sister-services, the Department of Defense, professional colleagues, and international partners. Due to security requirements for the AFPC Talent Management Data Environment (TMDE), the deployment of the chosen SAS Viya platform is delayed beyond the confines of SPARK's Phase II time parameters. Consequently, the focus of the current effort shifted to the preparation for Federal Risk and Authorization Management Program (FedRAMP) Impact Level 4 (IL-4) authorization to operate, a security level AFPC/DSYX will also require. Furthermore, SPARK Phase III was implemented during Phase II, which entailed using a Freeware Pascal compiler for Windows 10 that is part of the "Lazarus" software package to compile the legacy SPARK 1.0 macros in Turbo Pascal mode. (The SPARK programs that would not compile had to be modified in format or logic.) Once all the SPARK macros post-2004 were compiled to run natively in a Windows 10 environment, the resulting SPARK 2.0 system could be used on the Air Force Network, in advance of the arrival of the upcoming SAS Viya developmental sandbox for the Common Analytical Platform for A9, AFRC, AETC/SAS, and the AFPC.

The following is a brief description of the AFPC/DSYX SPARK system. For a more comprehensive review, see Weissmuller and Cazares (2020).

2.1 SPARK Studies

To support the AFPC/DSYX mission cited above, SPARK studies are generally performed in two primary categories: Program Evaluation and Air Force Personnel Program Process Improvement.

Category 1: Program Evaluation

The first category is the exercise of due diligence in program evaluation by the Program Manager of the Air Force Examining Activities Program (EAP), (a.k.a. Chief of Air Force Testing and Survey Policy). This office budgets the funds paid to Air Force personnel in support of the United States Military Entrance Processing Command for all pre-accession data collection on Air Force applicants as mandated in Department of Defense (DoD) Instructions and the Air Force Manual (AFMAN) 36-2664, 16 May 2019. The purpose of the program evaluation is to ensure that the tests given at the Military Entrance Processing Station (MEPS) and Military Entrance Test Sites are providing the Air Force with proper value. Longitudinal studies are performed to determine how well those tests predict success both in the Air Force at-large (i.e., a selection issue) and for classification purposes into specific Air Force training programs and subsequent career

progression. Longitudinal studies are performed to evaluate the efficacy of the *test instruments and policies* for Air Force purposes, not as a measure of the success or failure of any given individual.

Category 2: Air Force Personnel Program Process Improvement

A second category of SPARK projects is for process improvement in Air Force personnel policies. These analysis projects span in scope from the broad Air Force-wide selection issues to targeted protocol development for screening candidates for classification of new recruits into specific Air Force Specialty Codes (AFSCs) or post-accession assignment decisions into special duty identifier roles, such as First Sergeant or Military Training Instructor. Examples of these studies include:

1) Creating large benchmark "applicant pool" datasets representing the personal attributes of the "qualified" Air Force applicants in various fiscal years;

2) Benchmarking "learning difficulty" for all Air Force career field training pipelines to help rationalize differential career field entrance requirements;

3) Realigning AFSC entrance requirements to minimize adverse impacts without a loss of operational readiness;

4) Recalibrating cognitive standards for combat career fields formerly closed to women;

5) Introducing Predictive Success Models (PSMs) (i.e., mathematical composite scores), measuring the quality of *person-job-match* by utilizing the following measures:

   a) Cognitive (e.g., aptitude test scores);

   b) Non-cognitive (e.g., personality, job interest, Air Force Core Value Compatibility);

6) Developing and deploying structured interview processes, such as the 360-protocol used for selecting Military Training Instructors and other positions.

2.1.1. Related Scientific Professional Interactions and Studies

Incidental to the above categories, reporting results in both official and professional venues is also accomplished. When proposed studies/papers are not in direct support of an AFPC/DSYX mission and they plan to use Air Force data, they are vetted through official channels and abide by the relevant Institutional Review Board and Protection of Human Subject protocols.

The AFPC/DSYX serves in direct support of the USAF EAP per AFMAN 36-2664, 16 May 2019: *The Air Force Personnel Assessment Program*. The EAP is functionally located in AF/A1PT, Air Staff Directorate Force Management Policy for Accessions and Training. "Examining Activities" is a standard line item in the Air Force section of the DoD budget, directly adjacent to "Recruiting". In this capacity, AFPC/DSYX inherited the 50 years of broad Air Force level personnel research missions formerly supported by the AFHRL and its predecessor organizations.

The Program Manager for the Air Force EAP is also the central manager for the Air Force Strategic Personnel Research Council, as established in AFMAN 36-2664. This function is called the Air Force Center for Applied Personnel Studies (AFCAPS) and includes input from AFPC/DSYX, the Air Force Research Laboratory, Air Force Recruiting Service (AFRS), the USAF Occupational Analysis (OA) Program, the USAF Airman Advancement Program, etc. (See the USAF EAP, the AFCAPS and associated organizations in Figure 1 below. Note that some of these generic organizational labels include the other US Military branches (Army, Navy, Marine Corps, and Coast Guard), both directly and indirectly through the DoD level Under Secretary of Defense for Manpower Accession Policy Working Group. International connections include the International Military Testing Association, which generates many individual allied interactions as well as through various international committees. Professional organizations include the Society for Industrial and Organizational Psychology, the Performance Testing Council, and others. These contacts also generate professional studies on relevant topics of common interest.

Figure 1. Examining Activities and AFCAPS

2.2. Overview of the SPARK System

AFPC/DSYX manages the SPARK databank to support the research, development, validation and monitoring of USAF-wide personnel measurement, selection, classification, utilization, promotion, and proficiency evaluation processes. SPARK resources support improvements in Air Force human resources management processes while maximizing the return on investment and maintaining compliance with AF policies, professional standards, and strategic human resource management objectives.

To accomplish this mission, the SPARK system is structured into three components: the SPARK Databank, the SPARK Multi-media Library, and the SPARK Software Resource Center.

2.2.1. The SPARK Databank

In order to create a Predictive Success Model for every Air Force career field, the SPARK Databank includes both person attribute data and job requirements data. "Person" attribute data includes pre-accession test scores, job interest ratings, success in training outcomes, career progression history, job satisfaction self-reports, and re-enlistment/attrition records. The SPARK Databank also includes job requirements data about each Air Force Specialty (structure, job-interest profiles, literacy requirements, physical demands, and learning ability requirements).

At its core, the SPARK Databank has integrated all Air Force historical data archives (formerly known as the HRRD), with 20 additional years of ongoing routine data-pulls from Systems of Record (SORs) from the Air Force, DoD, and other services. As the action-arm of the USAF Examining Activities Program, the PhD staff of AFPC/DSYX monitors both the current functioning of all Air Force personnel tests (i.e., program evaluation for tests in the Air Force Personnel Test (AFPT) Catalog) as well as monitoring the state-of-the-art in I/O psychology to propose and evaluate strategic investment opportunities to improve operational Air Force personnel programs.

To support this professional analysis mission, the SPARK Team within AFPC/DSYX must obtain and securely manage all possible sources of available data regarding USAF applicants and current members to produce an array of studies and analyses in support of the development, revision, and validation of assessment screening tools and processes.

The SPARK team maintains ongoing and ad hoc data submissions from current Air Force and DoD sources, special data collection contracts, and historical data, providing access to records from data files that span over more than 50 years. The kind of data maintained by the SPARK Team includes contractor-collected data from special projects, survey efforts, enlisted technical training data, AFSC job/task analysis data, aircrew training data, officer training data, and new data to be incorporated as required by HQ AF Force Management Division (HQ AF/A1PF) in support of

future Force Management Policy analyses.

There are many sources of data, to include the following: AFRS, Basic Military Training (BMT), the Air Education and Training Command's (AETC's) technical training, flight training records, job performance, and the emergent AETC training ecosystem. The goal of the AFPC/DSYX SPARK team is to obtain, audit, prepare in usable form, maintain, and document the necessary data required for analyses to support the Examining Activity's function in the U.S. Air Force as a whole.

To that end, the SPARK team maintains the SPARK system, which is a highly indexed and cross-linked databank containing current as well as historical and longitudinal databases relevant to human resource policy topics. These policy areas include accessions, testing, personnel, performance, classification, AFSC task analyses, and training data used to support studies and analyses of issues impacting recruiting, accessions, personnel measurement, selection and classification systems, diversity, fairness, force development, force quality, utilization, and retention.

2.2.2. The SPARK Multi-Media Library

The SPARK "knowledgebase" contains over 2,000 technical reports (scanned in their entirety and key-word indexed) that document the personnel research methods and studies performed by the AFHRL with its predecessors and successors. SPARK also includes documents from other services and the DoD, including multiple reports from Air Staff archives as well as their associated datasets (e.g., assessments across all AFSCs of strength-aptitude requirements, literacy requirements, and other individual OA studies within each Job Family).

Other historical resources include hundreds of microfiches with archived Occupational Reports on Air Force Specialties, several hundred CD-ROMS indexing and archiving past research studies of the AFHRL from 1957—1998, (with limited hard-copy study-folders), and the library of professional personnel research books donated by the Institute for Job and Occupational Analysis.

In addition to the historical archives, the SPARK Multi-Media Library includes current AFPC/DSYX contracted and in-house research published in DTIC, under our imprint (AFCAPS reports).

SPARK Multi-Media Library resources also include a video library with military personnel programs (e.g., 1957 AF Video on Specialty Knowledge Test Development and 1961 AF Occupational Analysis/Personnel Research), as well as a growing number of realistic job previews and animation-based Situational Judgment Test items, circa 2018. The aforementioned resources are available on the AFPC/DSYX Restricted Network Drive or on the following sets of DVDs:

AFHRL_Technical_Reports, "Occeumetrics" DVD, briefings and other agency reports, Occupational_Research_Data_Bank, USAF_officer_research, and the remotely-piloted- aircraft-and-pilot_selection_development (limited distribution).

2.2.3. SPARK: Software Package and Research and Development (R&D) Center

The Applied Performance Assessment and Testing (APAT) Center is a psychometric laboratory established by AFPC/DSYX for the controlled development and use of experimental (i.e., *psychometric*) data. AFPC/DSYX collects data through the APAT center in order to create and validate replacement assessments as well as to develop new assessment instruments. Once the data flows to a designated psychometric SPARK Job Family, the dataset is deemed "experimental" and per the Informed Consent Form established with APAT, never becomes part of a subject's personnel record or used in any official Air Force action. To wit, the AFPC/DSYX only releases *psychometric* data for policy evaluation purposes and model development, in accordance with established official agreements (e.g., DUA, Memorandum of Understanding (MOU), Memorandum of Agreement, etc.) and in the case of APAT, the Informed Consent Form.

The *operational* "personnel records" imported into SPARK are maintained solely for the statistical research or program evaluation purposes of the program manager of the Strategic Personnel Assessment Program (SPAP) (HQ AF/A1PT Examining Activities and the SPAP per AFMAN 36-2664). As a responsible steward of Air Force personnel data, AFPC/DSYX's data management plan ensures program effectiveness and compliance with all selection, classification, and professional testing standards as well as protection of personally identifiable information (PII). An initial step in importing personnel data into SPARK is to remove PII and assign each *statistical record* a SPARK-ID capable of tracking and matching records across decades, regardless of original ID tracking (i.e., SSAN, DoD ID/EDIPI, Air Force Serial Number, Air Force Applicant ID, etc.). Once incorporated into the SPARK Databank from any external SOR, the data becomes part of the databank of statistical observations, subject to modification only for statistical analysis purposes, and can no longer be released or used to make decisions on the rights, benefits, or entitlements of any individual. This is in accordance with the AFI 33-332, para 2.14.4.5 Privacy Act (k)(4) exemption which "…applies to records maintained solely for the statistical research or program evaluation purposes and which are not used to make decisions on the rights, benefits, or entitlements of an individual".

The primary focus of the current SPARK development project is the set of software and procedures used to maintain the SPARK Databank. The goal is to upgrade all these processes (anchored in the 1960s) to 21st century methods, while at the same time, maintaining the flexibility of a "databank" philosophy to index and retrieve datasets from diverse sources, each with unique challenges.

The portal to SPARK is the set of algorithms used for running SPARK programs, which have been

compiled during this Phase II from the original Turbo Pascal programming language to a format that can run natively in a Windows 10 environment on the Air Force network to support AFPC/DSYX's psychometric and operational requirements in the near and more distant future. The updated macros include compiled utility, layout, code-generation, encryption, and special-purpose programs. The foundation of the SPARK databank is the Data Job Families that were first annotated in the SPARK I system documentation and subsequently updated and submitted as an AFCAPS document (Weissmuller, Yi, & Cazares, 2021) as a companion to this SPARK II report. The updated and restructured SPARK macros and the current Data Job Families serve as the basis for all analyses by the AFPC/DSYX.

3.0 SPARK SYSTEM AND WORKFLOW MANAGEMENT

This documentation describes the SPARK system, including History, Management, Design Philosophy as well as technical descriptions for external data imports, existing historical (Master File/Project File) data sources, and data capture procedures for internally or experimentally generated datasets. Also included are the indices of the archive of historical technical reports and documentation of the Job Family method for file cleaning and integration processes, file storage, timelines and frequencies of pulls and updates as well as the file structure for each data source.

3.1 SPARK Databank History, Management, and Design Philosophy

The SPARK Databank has incorporated Air Force historical databases from over 50 years ago to the present day, and continues to integrate new data streams as new sources and assessment instruments are developed. Some "people" datasets are deemed to be "predictor" data, such as personnel pre-accession test scores, job interest profiles, ability, personality assessments, etc. Other "people" datasets are "criterion" data, such as success in training, job satisfaction, and retention decisions. The other kind of data (i.e., "job" data), includes data collected to measure AFSC requirements, such as literacy (e.g., reading grade levels and need for specialized domain jargon), physical demands (e.g., strength and stamina), and learning difficulty demands from each training course pipeline. The SPARK resource services data requests from a diversity of organizations and individuals: AFPC/DSYX's industrial/organizational psychologists, Career Field Managers (CFMs), the Air Force Institute of Technology (AFIT), the RAND Corporation's Project Air Force (PAF), and from a variety of other Air Force functionals and multi-level Working Groups. All of the aforementioned types of datasets must be in a format that facilitates rapid extraction into high priority projects, with a minimum "data cleaning" load for any requesting project analyst.

In accordance with the historical data management principles and Findable, Accessible, Interoperable, and Re-usable (FAIR) guidelines, AFPC/DSYX developed both a Project Charter Protocol to scope emerging projects and a SPARK Data Pull Request Form to instantiate the data to

be used in the project. The Project Charter documents the objective for a given project along with the contact information for all the Stakeholders and staff members working on the project. The SPARK Data Pull Request Form is completed by the AFPC/DSYX project's Principal Investigator (a.k.a., Task Scientist) and defines the kinds of data and date ranges to be covered. The SPARK technician will translate this into the actual datasets required as well as the dataset merge required to produce a de-identified project data file. The actual datasets used by the SPARK technician will provide the project analyst/principal investigator with references to the job family (with explanatory information), and layout (data dictionary) with the variable identification, format specification, and meanings for each value of coded variables. This data management practice facilitates the "finding" and "accessing" of SPARK data by research clientele.

3.1.1. Translating the HRRD to SPARK Program Management Models

Because SPARK incorporates the 30 years (1968-1998) of data and procedures from the Human Resources Research Databank that was managed by the AFHRL, any indices of these older resources use the then current role titles for personnel accomplishing the research program. Note that this SPARK Team is a subset of AFPC/DSYX. In function, however, AFPC/DSYX is attempting to replicate the functionality of two divisions of AFHRL, specifically, the Manpower and Personnel Research Division and the Technical Services Division. The Technical Services Division aggregated decades of data into longitudinal databases and provided analysts to execute research projects requested by Task Scientists in the Manpower and Personnel Research Division.

Figure 2 shows the organizational breakout within each of these two aforementioned divisions. The SPARK team provides the technical services indicated in the figure, except for the actual main frame computer (Computer Operations) and other hardware management (Information Resources). Not necessarily clear from the figure, the Computer Programming branch included the units dedicated for collecting and archiving Master File data from all external sources as well as managing the layouts and code structures as they evolved over the decades. The work associated with layouts and code-tracking (File Item Data Overview, FIDO) will be discussed in depth late. The Computer Programming branch also included several units dedicated to doing data analysis per task scientist requests. These project descriptions may have required the assigned analyst in the branch to translate notional data sources into exact available data sources (using a Generalized Data Index Table, (GDIT)), layouts, and FIDO code structures in order to create the dataset appropriate for the projects as described in the project request. Translation of these legacy systems (GDIT, FIDO, etc.) will be addressed in detail in sections that follow.

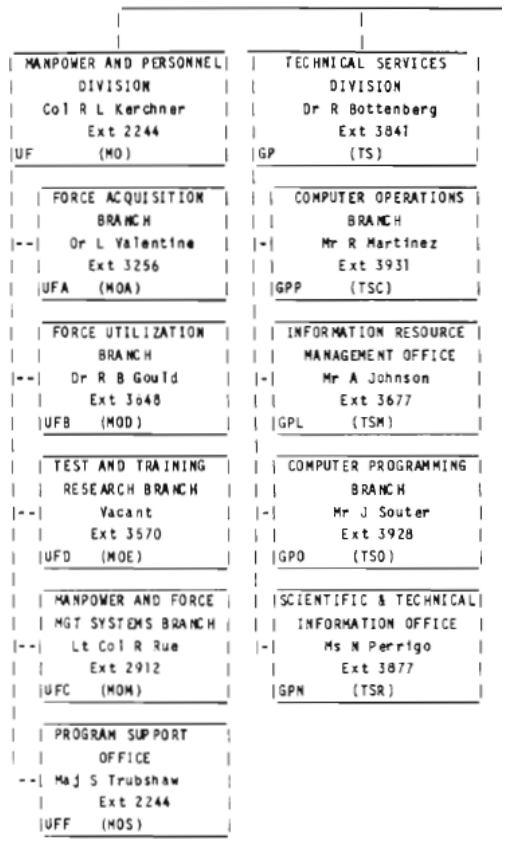


Figure 2. AFHRL Divisions in 1984

Within AFPC/DSYX, the I/O psychologist staff provides the functionality of the AFHRL Manpower and Personnel Research Division. Figure 3 provides the 1985 Thrusts for this AFHRL Division:

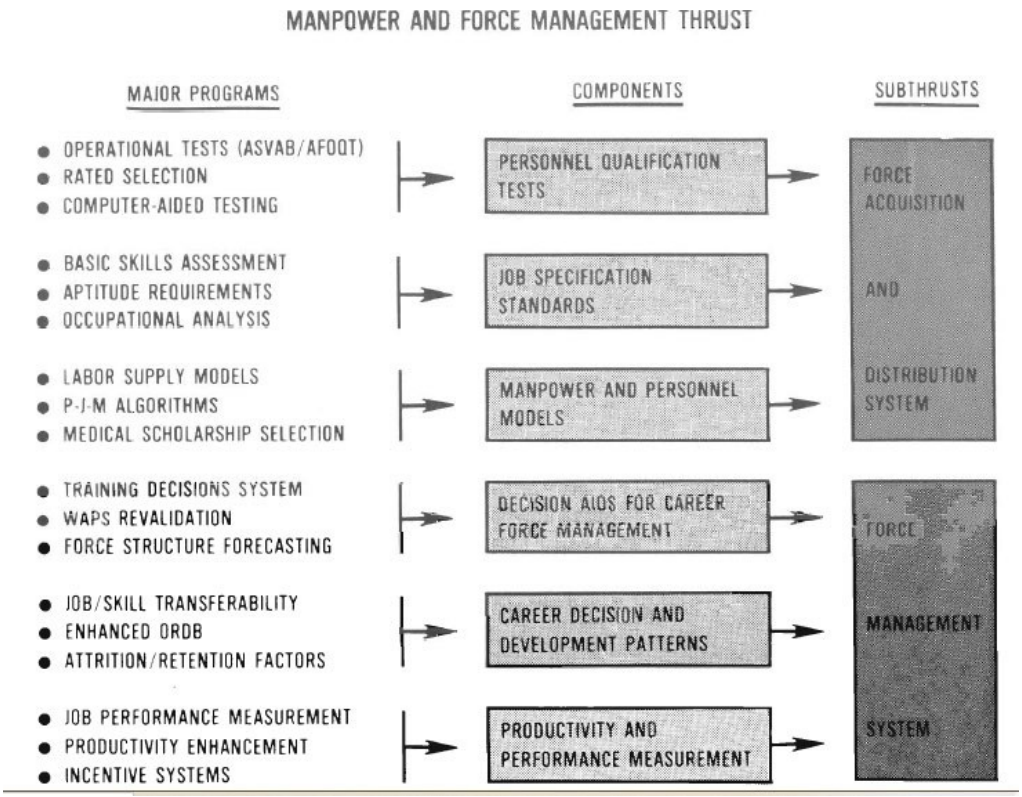


Figure 3. Manpower & Personnel Research Thrusts 1985

Below is a list of the new SPARK role titles that correspond to each identified AFHRL-period's equivalent job position:

3.1.1.1 SPARK Program Manager – AFHRL Computer Programming branch chief in the Technical Services Division. This person manages interactions with external agencies for establishing Data Use Agreements (or Data Sharing Agreements, or Test Use Agreements [TUA]), as well as Privacy Impact Assessments (PIAs), System of Record Notices (SORNs), and any other Air Force, DoD, or federal requirements for disclosures of data maintained by the Government. All such documents are prepared for signature by AFHRL Division management (currently, AFPC/DSY division chief). The SPARK program manager reviews proposed projects/studies, clarifies requirements and suggested analysis needs, and then assigns projects to the appropriate analyst(s). This Project Review is comparable to the Project Charter Review currently conducted by the AFPC/DSYX branch chief. The focus is only on the SPARK team and therefore only involves review of the "SPARK Request Form", discussed later in this report.

3.1.1.2 SPARK Data Importer – AFHRL Member of the Computer Programming Branch who requires specialized training and certifications to directly (or indirectly) access data from

our external sources as authorized in a DUA. Currently, when these data are pulled from their external source, the new file is stored on the AFPC/DSYX restricted drive (DATA-TRANSER directory) and the SPARK Data Importer alerts the SPARK Data Manager (described below) to the new submission. In today's AFPC this is done by AFPC/DSYR (Data Retrievals) for *operational* data sources and AFPC/DSYX for a combination of *operational* and *psychometric* data.

3.1.1.3 SPARK Data Manager – AFHRL Unit Team Leader in the Data Management Unit. Due to the new PII-handling requirements for SPARK, this person may be the POC for all first- level data importing, up through the removal of PII from imported datasets.

3.1.1.4 SPARK Job Data Family Specialist – Member of the Data Management Branch with responsibility for importing, normalizing, sanitizing, and updating Master Files within their Job Family areas. This person's focus is to update a Master File for subsequent use by other analysts responding to Project Charter needs.

3.1.1.5 SPARK Data Analyst – AFHRL System-Analyst/Programmer in the Computer Programming branch running "study" projects as assigned in a Study Package written by the AFHRL Task Scientist (now aa AFPC/DSYX staff psychologist, typically a PhD I/O psychologist) and approved by the AFHRL branch chief. This person translates the Project Charter and SPARK Data Request into actionable analyses. Note: Due to the analytical expertise and computational prowess of today's staff psychologists, often times the SPARK data analyst simply turns over the extracted custom SPARK project dataset to the requestor.

3.1.1.6 AFPC/DSYX staff psychologist – the AFHRL task scientist working in the Manpower and Personnel Research Division. Many years ago, very few AFHRL task scientists did their own data analyses. In modern times, most I/O psychologists perform their own.

3.1.1.7 High-Level Sponsor – The USAF office or organization with a need for personnel research that will be translated into a research study package (currently called a project charter) by the AFHRL task scientist (now, AFPC/DSYX staff psychologist). In the past,this sponsor would write a Request for Personnel Research (RPR). A review panel consisting of the Technical Services Division Chief, the Computer Programming Branch Chief, and the USAF Examining Activities program manager would meet to establish the priorities for these RPRs over the next 18 months. (In accordance with AFMAN 36-2664 (16 MAY 2019), the Strategic Personnel Research Planning Council identifies prioritized areas of personnel selection and classification concerns for all officer and enlisted Air Force Specialities and then provides recommendations to the Chief of the Strategic Personnel Assessment Program to be included in the research agenda annotated in the Strategic Personnel Research Plan (Section 3C—Program Strategic Planning).)

Currently, the SPARK Program Manager (PM) has inherited and adapted an ad hoc data management workflow with an adherence to FAIR principles of data management. In practice, AFPC/DSYX's SPARK databank is managed to support applied research purposes and outputs that can be evaluated and reused: SPARK data is managed by AFPC/DSYX to be found, accessed, reformatted, and reused by the USAF managers and researchers (cited above), who often request that the SPARK PM and/or staff psychologists develop PSMs for them, rather than receiving raw data extracts. By performing the requisite analysis and subsequent synthesis of a working model, DSYX provides process-improvement in support of AFMAN 36-2664, Personnel Assessment.

3.1.2. Data Classes: Operational, Psychometric, and Job Requirements Data

In concert with the principles of FAIR-ness, the current SPARK data management plan is guided by the Department of Defense Instruction (DoDI) 3200.12: DoD Scientific and Technical Information Program (STIP), 22 AUG 2013, yet with qualified exemptions. There are three classes of data that comprise the SPARK databank: *operational* data drawn from a SOR and *psychometric* data generated for development of new assessment instruments or data generated for revision of existing ones. As for the operational data, AFPC/DSYX is covered by the following exemption: DoDI 3200.12 does NOT apply to "DoD programs involving day-to- day operations the warfighter uses unless required for scientific and technical analysis." If dissemination is required of operational data, AFPC/DSYX refers the requesting agency to the originator of the dataset. As for psychometric data, AFPC/DSYX follows professional guidelines tailored to each work product by disseminating data to the highest degree possible. For example, in the case of a report that contains Controlled Test Material (CTM), AFPC/DSYX adheres to professional guidelines by publishing a Restricted Distribution document. AFPC/DSYX may follow that publication with a redacted version (sans CTM) that is published with Unlimited Distribution. AFPC/DSYX only releases psychometric data for policy evaluation purposes and model development, in accordance with established DUAs.

SPARK data is managed for "interoperability" and re-use, since information-requestors are able to reformat the data in accordance with their particular purpose; for example, in the following possible formats: .CSV (text file using a special delimiter character), .TXT (text file with fixed field widths), .XLS (Excel spreadsheet), or .SAS (Statistical Analysis Software dataset). SPARK procedures/macros exist to generate load scripts for other analysis tools such as the Statistical Package for the Social Sciences (SPSS), which is used at the Air Force Research Laboratory (AFRL) and by several contracting organizations used by AFPC/DSYX.

In summary, all documentation and dissemination of information from the SPARK databank is *informed* by DoDI 3200.12, as well as each unique DUA or MOU between AFPC/DSYX and partnering agencies. AFPC/DSYX endeavors to abide by best practices in data management at all times and as infrastructure and instruments (hardware, software, applications) evolve, the SPARK data management plan is modified in accordance with

the aforementioned guidelines, to support the new, ever-changing model. Though the data management procedures evolve with new requirements, the AFPC/DSYX mission remains steadfast: To facilitate USAF total force capability and to optimize mission readiness via support of the strategic policy analyses by the HQ AF/A1PT Examining Activities program.

3.2 SPARK-ID and Protecting Personally Identifiable Information (PII)

Like SPARK's predecessor, the Air Force HRRD (1968-1998), every "personnel" SPARK Master File datum is stored in a flat American Standard Code for Information Interchange (ASCII) format. This means any data file can be inspected and easily read by humans with a NotePad or Text-based Editor. However, unlike the HRRD (which used a Social Security Account Number (SSAN) as the master key for all personnel records), SPARK uses its own uniquely generated identifier called the SPARK-ID. For maximum backwards compatibility, a SPARK-ID is a 9-character alpha-numeric value (the same size field as a Social Security Account Number), yet this unique identifier is not a simple encryption of the subject's SSAN. Moreover, unlike the SSAN, the SPARK-ID is not considered PII.

Upon receipt by the SPARK data manager, all data records from external organizations are moved to the secure off-network SPARK clean-room. The external SOURCE files are put through the appropriate import process. When the import is completed, the original SOURCE data file is archived in the secure off-network SPARK clean-room, in case reprocessing is required. Preparation of a releasable SPARK data file (i.e., a file which may leave the SPARK clean-room) has all SSANs, DoD IDs (formerly called Electronic Data Interchange Personal Identifiers (EDIPIs)) or any other PII (as well as other highly unique fields) removed, leaving only a single 9-character unique SPARK-ID. Standard PII removed (by label and known content) include: SSAN, DoD ID, Air Force Serial Number, and Recruiting Service Applicant ID. Other variables discovered in the standard import protocol with a high percent of unique values are also removed. These non-standard variables have included such items as Student ID, Unit Manpower Document (UMD) Position, and a variable labeled "ASSIGNMENT", which actually contains the 11-digit SSAN with dashes.

Once processed into the SPARK Databank, these records are only a statistical snapshot in time, not a personal record to be kept current or updated or replaced by newer data. Since these data are statistical observations, it is prohibited to pull individual records for the purposes of using that data within for any personnel action or to determine the rights, benefits, or entitlements of any individual. In SPARK operations, records are never individually pulled by SSAN, DoD ID or any other form of PII. Furthermore, PII and highly unique data elements have already been removed during the "import" protocol, long before release from the Clean-Room for any analysis. In operation, SPARK records are pulled from a SPARK master file because of characteristics (variables) relevant in the aggregate to the Air Force policy being evaluated, such as "success of anyone who took a specified training course in 2017-2019."

3.2.1. SPARK-ID to Support Cross-Master-File Record Matching

To be useful for policy analysis, every record in a releasable SPARK file must have a "SPARK-System-wide" unique SPARK-ID. For example, in any statistical analysis project, this requirement ensures that the predictor variables (such as pre-accession test scores) from one SPARK Job Family (0843) can be matched to their corresponding criterion value' for example, final school grade for the training course in a different SPARK Job Family (0801). These paired records are merged (based on a common SPARK-ID) into a single record for each person in the entire set of people in a specified training course. If some records do not find a match between the two datasets, they are dropped from the analysis. This analysis is not about individual performance, but instead, a search for a statistical relationship such as that between pre-accession test scores (a policy set prerequisite) and success in a training course. The set of matched and merged scores are run in a regression analysis to create an evidenced-based PSM. PSMs with a minimum passing score will then become codified in the Air Force Enlisted Classification Directory as a new additional requirement. This new classification requirement will screen future applicants for entry into that AFS with that training course. This de-identified dataset used to create the PSM will be archived to document the basis for that new personnel policy for that particular AFS. No product from this developmental analysis will ever go back into any record of the unknown individual's, whose de-identified records were used in an evaluation that culminated in policy. In most cases, as this is evidenced-based policy analysis from the toolbox of I/O psychologists, in order to be pulled for inclusion in this study, each subject/student must have already been selected for and finished the target course, and has therefore, passed the stage of their career for potentially being affected by any new resultant policy change.

3.2.2. SPARK-ID Management Logistics

Although the SPARK-ID is a 9-character alphanumeric string, the unique identifier is not simply an encoded form of the associated person's SSAN. The first encounter of any new dataset can cause the generation of the "base" SPARK-ID for people in that dataset, regardless of the PII available in the dataset. Hence, if the DOD ID is available on that dataset, the *SPARK Identify Master File (SIMF)* will be searched for a matching DoD ID. If no match is found, the next available randomly generated SPARK-ID will be used, which also encrypts the flag that the source was a DoD ID. The possibility exists that an individual may have multiple SPARK-IDs until a dataset enters SPARK, which actually necessitates two or more PIIelements to permit the "cross-walk". Every Master File import is processed via the SIMF, at which time the SPARK-IDs are compared and the "original" SPARK-ID is propagated to the new Master Data File stored and indexed in the SIMF.

The SPARK Databank manages (oversees, indexes, imports, and integrates) a large collection of

diverse datasets from many outside sources, each with its own requirements, quirks, and history of changes in operation. (This process is explained later in the discussion on Job Families and their unique scripts for importing.) Regardless of its original database host system, data imported into SPARK generates a Master File stored as a flat, ASCII text file. In most cases, the external dataset is exported from its original format in order to provide SPARK with a CSV (Comma Separated Value) format file with the variable labels stored in row one of the data file. In some cases, the external dataset is provided in a fixed length field for each data item. Fixed length record files also come with a separate file/document called a data dictionary (or file layout), which delineates specifications for each data field in the file. From either format, data are tentatively imported into SPARK and become a fixed length record with a fixed width for each data field, and then is documented by a data dictionary called a "layout." The SPARK version of a layout file is also a text file with one or more rows of specifications for each data field in the imported file. SPARK routines exist to accept the SPARK layout file, automatically produce a run-stream (batch file) to perform a data distribution on the imported file for each data field, inspect that field, and then recommend "reformat" actions and flag potential PII (and/or highly unique items) for an analyst's review.

As each imported dataset is extracted from SPARK, its indexing key (usually SSAN or DoD ID) is matched against the current SIMF to ensure the first or "base" SPARK-ID is used to maximize match rates between other SPARK statistical datasets in the current analyses. Aside from not being PII, another advantage of the SPARK-ID is the following: Before SPARK, if two databases did not contain exactly the same PII elements for matching, those records could not be combined for statistical analysis. In contrast, with the SPARK-ID, as soon as another dataset is imported with two or more varieties of PII elements, a bridge is forged, and now the two datasets with only unmatching "unique" PII can be matched and used in a statistical analysis.

For example, consider the pre-accession job interest values yielded from the Air Force Work Interest Navigator (AF-WIN). AF-WIN values are indexed by a person's unique USAF Recruiting Service Applicant ID, and there is no other index PII key available on that dataset. By itself, that file cannot be used for evaluation if the AF-WIN scores result in better job satisfaction (and is worth the cost of "job preference" matching to the Air Force). When the USAF Recruiting Service swears-in a person and then ships them to BMT, this action generates a record in the dataset (specifically, Job Family 0850-Recruit-RDB), which includes both the USAF Recruiting Service Applicant ID and their SSAN. At the time of import, the SIMF is triggered to incorporate the SSAN into the existing SIMF record for their/SPARK-ID. In this way, an applicant's responses to Job Interest (pre-accession) can be linked to a reported job satisfaction cited in an occupational survey taken after four years on the Air Force job. Another benefit to the SPARK-ID is that all people who took the AF-WIN yet who did not enter the Air Force can be "identified", because those candidates have a SPARK- ID with only an AFRS ID and no other PII. This differentiation allows the I/O psychologist to identify the

iinterest profiles of people who do not come into the Air Force. The subsequent AFPC/DSYX analyses can then be shared with the Recruiting Service Market Analysis shop.

The master SPARK Identity Master File (SIMF, 0999-SIMF.DAT) is maintained in the SPARK clean-room. Each SPARK-ID record in the SIMF lists the previously encountered personal identifiers linked to this SPARK-ID (i.e., SSAN, DoD ID, Air Force Serial Number, and Recruiting Service Applicant ID). The original version of the SIMF was generated from the Personnel Data System (PDS, Job Family 0501) using the DoD ID as the preferred foundation for the "base" SPARK-ID for each person. PDS is an extract maintained by the Air Force Personnel Center and is a cumulative archive of both enlisted and officers in the current force, as well as those who have served in the past. Other than the SMIF, only the submission from external data providers maintains their complete PII in the SPARK clean-room; however, these "AS SUBMITTED" files are not releasable to anyone other than the SPARK Data Manager.

3.3 SPARK "Job Data Family" Management Approach

To best support a variety of projects, the data handling portion of the SPARK system is structured as a "databank", as opposed to a single "database." To illustrate via an analogy, assume each data source (by organization) and "kind" of data is routinely collected and then stored within its own safety deposit box with its own access requirements. In SPARK, the safety deposit box is actually a computer directory labeled with a Job Data Family Number (typically called a "Job Family", for brevity). Because of differing rules of engagement, there must be differentiation between the three primary classes of Job Family data that are imported into SPARK. *Operational* data are data record "snapshots" drawn from an official SOR belonging to an outside organization. *Psychometric* data are data generated incidentally in the development process for new assessment instruments and/or for revision of existing instruments by IAFPC/DSYX /O psychologists. In other words, the generation of psychometric data for analysis by psychologists is akin to a "work product" for lawyers. *Job Requirements/Attributes* data are records that document attributes of Air Force jobs or occupations, as opposed to data about people. These measures for AFSCs cover topics such as strength-aptitude requirements, literacy reading- grade level, and task and job learning loads.

Each SPARK class has its own rules for the "release" of data. For example, AFPC/DSYX does not release *operational* data. Instead, it refers any requesting agency to the originator of the dataset to inquire about its release. AFPC/DSYX only releases *psychometric* data for policy evaluation purposes to qualified research teams in accordance with established DUAts. This de-iudentified data may also be released to duly authorized working-groups or action teams for statistical model development. *Job Requirements* data are rarely requested by users outside of the AFPC/DSYX professional staff.

The historical SPARK databank spans over 50 years and is comprised of Master Files, layout files, and other data files, all labeled for easy identification. The core SPARK databank of historical records is composed of records drawn from the official System of Record from the Air Force or from DoD sources. This databank includes several "Job Families" such as Accession Test Scores (Armed Services Vocational Aptitude Battery (ASVAB) (0843) from the Defense Manpower Data Center (DMCD), Tailored Adaptive Personality Assessment System (TAPAS) (0950) and Cyber Test (CT) (0960) from MEPS, AFRS "As Shipped to Basic Training" (0850) Data, Occupational Survey Data (0060) from Air Education and Training Command (AETC)-OA, Personnel Data System (PDS) Personnel (0501) Data from AFPC, Flying Training (0665) from AETC, Enlisted Technical Training (0801-ADSS) from AETC, Weighted Airmen Promotion System (0851) promotion scores, and Air Force Officer Qualifying Test (AFOQT) scores covering Job Families 0641, 0986, 0987, 0991, 0992, and 0993).

Prior to any data flow to a designated SPARK Job Family, the source agency establishes their System of Record repository system and then a DUA is negotiated establishing the data use (or sharing) relationship. All of these prior "records" are accrued to the aggregate databank of "statistical observations" in the designated SPARK Job Family, meaning that they are no longer deemed to be an official record on a person but rather an "example" statistical observation from a given point in time. As new observations are added, a transmission loss rate of 5% or less of incoming records is considered acceptable, i.e., within statistical limits. No attempt is made to replace earlier records with newer records for the same SSAN within any data family. AFPC/DSYX also analyzes "test—retest" dynamics, which means data is needed from all attempts at a given test to determine the stability of the test over time. Furthermore, these statistical observations may be changed to improve the quality of the data for a given analysis. AFPC/DSYX is also responsible for doing adverse impact studies, so if a record is missing demographic information such as gender, race, or ethnicity, it may be modified based on matching that data from other data sources which have that same data element.

These records are maintained solely for the statistical research or program evaluation purposes of the Program Manager of the Strategic Personnel Assessment Program (SPAP). To build on the analogy offered previously, each Job Family can be considered a different safety deposit box in the data "bank", with a compartmentalized set of rules and regulations for access and use of the data within. This compartmentalization of the data is necessary because each imported dataset has its own unique set of protocols (and challenges) both for importing and later extraction for use in model building or other statistical analyses. Moreover, SPARK collects then imports, de-identifies, normalizes, and merges longitudinal data flowing in from a highly diverse set of operational and research organizations, each of which are consolidating data from numerous remote offices. Some data sources have known issues, like data originating on mark-sense answer sheets which reads the black marks in bubbled areas. Responding on bubble sheets is subject to

human error in key fields like SSAN, date of test, or the person's name being bubbled and scanned as "RQY" instead of "ROY".

SPARK is intended to maintain these large scale, longitudinal datasets in an ever ready, interoperable state to minimize any delay in responding to data requests from AFPC/DSYX staff psychologists who, in turn, respond to analysis requests from Air Force managers. This means that one group of people in the SPARK team deal with external agencies to secure and integrate snapshots of these operational datasets on a recurring basis. The goal is to minimize the workload and schedule delays for AFPC/DSYX project analysts, who must use data from one or more Job Families to complete their project. Datasets are extracted from their Job Family (computer directory) then "matched and merged" into a de-identified project dataset (a.k.a. a database) only when the requirements of a given project are clearly defined. Because the project analyst is not the person managing this data on a daily basis, the Job Family structure approach allows the SPARK data manager to document all the quirks of each dataset (in the layout for that Job Family) to ensure the using analyst has access to all the rules of engagement with that data source.

Because SPARK Job Families cover datasets from both long-running operational programs and quick changing psychometric data from research programs, the SPARK data manager must vet every dataset turned over for import into the SPARK system. Some Job Families are easy to maintain, such as Job Family 0664 – TBAS (Test of Basic Aviation Skills) and Pilot Candidate Selection Method (PCSM). The primary score extracted from this Job Family is the PCSM score (soon to be renamed the PSM-11x score) for screening pilot and other aircrew applicants. What makes this SPARK import "easy" is that the PCSM contractor always provides a complete copy of their database, which only goes back to 2007. (This database has less than 43,000 cases as of 01 Jul 2020.) In other words, there is no pre-existing Master File already in SPARK to which the new data needs to be integrated. All data provided (after PII removal) and conversion of dates to four-digit year formats will be made available upon the next release of the SPARK Master File for this Job Family (0664-20E.DAT).

Many SPARK source files are from long-running operational programs for which there are existing Master Files, going back decades. Operational organizations sometimes upgrade their mission and procedures (i.e., computer systems or software packages), which causes change in the data they collect and hence provide to the SPARK Team. Rarely will a source agency provide a "heads up" summary on the nature or scope of changes. Often the source agency just processes its standing order (i.e., re-runs the existing script) for the SPARK data-pull. A common scenario is for the source agency to just "give you what we got", which requires the SPARK data manager to detect differences between the old standard and the new data submitted. Once the differences are identified, the SPARK data manager needs to decide what it means to add the new dataset to the existing longitudinal master file system. Sometimes, the master layout is adjusted to include data fields from both the old and new standards with corresponding annotations. Some variables

may no longer be provided (like AGE, for example, noted "no longer available since Jan 2020") and others may have their coding standards changed (such as RACE from 6 categories to 31 categories, generating two variables "RACEOLD" and "RACENEW" with the appropriate annotations in Layout comments as well as the explicit codes listed). Whatever the SPARK Data Manager decides will be documented in the COMMENTS and NOTES in the new Master File Layout. In some cases, the changes may be so dramatic that the existing Job Family designation will be declared to be a Closed Archive, and an entire new Job Family will be created to accommodate very different import protocols from this point forward. An example of this major disconnect is the conversion from AFOQT Form S (Job Family 0987, Aug 2005-Feb 2015) into AFOQT Form T, with three new Job Families (0991, 0992, and 0993, Feb 2015-present).

At times, there are separate Job Families dedicated importing highly unique raw data formats, yet that produce common scores that can be combined into a single database and exported for operational use from a different Job Family, which is the case for the AFOQT. This is an exceptional case, since AFPC/DSYX is responsible for the development and deployment of the AFOQT. AFPC/DSYX developed the pre-processing software to scan and score the AFOQTs, and then gave the software to AFPC/DP1 for operational usage under a different Job Family. The AFOQT test forms (Form S and Form T) later had unique demands, since the answer sheets were processed by the mark-sense scanner: Even though the tests appear different during scanning (i.e., one S answer sheet AFPT 987 versus two T answer sheets, AFPT 991 and AFPT 993), there are a small set of final AFOQT composite scores that are common among these tests, yet the ultimate use by the analyst can be delivered in an integrated format.

See the companion document to this SPARK II report (pending publication with the entitled "Strategic Policy Analysis Resource and Knowledgebase: SPARK Data Job Families as of September 2021" (Weissmuller, Yi, & Cazares, 2021) for the most current listing of the AFPC/DSYX's SPARK Data Job Families.

4.0 AFPC/DSYX'S SPARK TEAM'S MISSION, PURPOSE, & FUNCTION

The SPARK team is tasked with updating historical and current files on various data sources and DUAs. Further, the common functions of AFPC/DSYX consist of auditing the data files pulled/received, creating layouts for data files when needed, creating/updating Master Files for Job Families as applicable, maintaining the Data Item Code Explanations (DICEs)—(formerly, FIDO, which is the code file on all variables)—and creating/updating the Data Retrieval Item Locator System (DRILS), (formerly, the GDIT) Master File as well as the DRILS Roster. In addition to these routine functions, AFPC/DSYX is responsible for pulling data on an as needed basis for projects and analyses to support the Air Force Research and Examining Activity's requirements.

4.1 Daily Functions of the SPARK Team within DSYX

- Continuously pulling/receiving data based on the agreements with internal and external agencies.
- Auditing new data to make sure it is usable and can be incorporated with the previous submissions of similar data within AFPC/DSYX.
- Updating/creating appropriate files to incorporate new data pulled/received.
- Updating and maintaining appropriate Master Files within AFPC/DSYX.
- Copying active AFPC/DSYX Master Files onto DVD/CD-ROMs for internal and offsite storage and backup of the internal AFPC/DSYX server.
- Negotiate new MOUs with outside agencies as needed to receive/pull necessary datasets in support of AF research.
- Pull data and appropriate layouts of data as needed for AFPC/DYSX's internal/external use.

4.2 Overall Function/Goal of the DSYX SPARK Team

The SPARK team's function is to provide both quick and long-term internal and external/contracted research and analysis data support. This means AFPC/DSYX will either maintain within SPARK or have access to all data sources required to support force management analyses and needs.

5.0 TRANSITION FROM SPARK PHASE I TO SPARK PHASE II

SPARK Phase II uses the research and documentation from SPARK Phase I as a springboard to accomplish the following:

- Investigate the efficacy of the current SAS 9.4 and SPSS applications and any opportunities to integrate with open-source solutions.
- Research, evaluate, and recommend the way forward to replace existing SPARK processes with new programming and hosting capabilities.
- Identify/recommend technologies that improve coordination/integration for people/ processes.
- Facilitate data importation and integration for AFPC/DSYX data.
- Investigate automating SPARK system, to include possible adoption of Machine Learning/AI technologies

This second stage of SPARK was conceived originally as the design phase for the new programming and hosting capabilities, which would facilitate the importation and integration for the Strategic Research and Assessment branch's data to support their analytic research programs and efforts to market (brief) and implement recommendations for operational deployments and ongoing monitoring programs. The code production that would be required to modernize the SPARK 1.0 system and fulfill the Information Assurance requirements by getting the database fully on the Air Force network was slated for Phase III; however, the security issue became a significant

source of risk for AFPC/DSYX, causing a significant leftward shift in the project's timeline. Consequently, getting SPARK data on the mainframes became a primary goal for Phase II, in order to completely eliminate the need for stand-alone machines and for USBs that were used to port data from the offline scanner. Furthermore, the 16-bit SPARK software run on the offline XP machines or XP-emulators were connected to USB drives, which were required to store the large volume of SPARK data. SPARK files could not be on the Air Force network until they could be processed there. The leftward shift in the SPARK II project's timeline caused by the aforementioned security concerns eliminated the need for a Phase III and the three-year effort was completed in only two years. The following is a catalog of the SPARK effort, before and after the transition to Phase II.

The original security mandate for Phase II was to create Action Plan with milestones to ensure the AFPC/DSYX legacy platform is IA compliant. The SPARK software was outdated and the 16-bit Turbo Pascal programs (the macros) would need to be recoded to get them on a networked computer. The original concept was to recode the macros in R, to preclude the need for using offline computers and XP emulators to run 16-bit code. Aside from enhancing security by no longer using stand-alone computers and external hard drives to store SPARK data, the other benefit would be that the (currently) offline resources would be available on the network so that customers can access them. The second prong to the plan was to set milestones for converting all PII to a randomized SPARK-ID, which had already been included in the planning process. As mentioned previously, Phase I entailed the documentation of the SPARK 1.0 system; yet, the SPARK-ID and the PII purging process was still being documented and developed during Phase II. The SPARK-ID became necessary because applicant data is needed to conduct longitudinal studies and safeguards must be applied to protect the privacy of personal information, in accordance with the Government-wide SORN for personnel research (i.e., OPM SORN 6). So, a 9-character alpha-numeric randomized identification code called the SPARK-ID was developed that can track every person in the database, whether the person is listed with a social security number Electronic Data Interchange Personal Identifier (EDIPI) number (which is just the 10-digit number on the back of a CAC card), the Air Force Serial Number, or the AF Applicant ID. The aforementioned examples of PII are converted to a randomized SPARK-ID, which is NOT considered PII. The challenge during Phase II is that the SPARK-ID is written in 16-bit code (the MS-DOS version of Turbo Pascal programs) that can only run on Windows XP machines or on host machines with XP-emulators, so originally the plan was to convert the SPARK-ID algorithm to R code during Phase III. Instead, the SPARK-ID and SPARK-ID—related programs were updated during Phase II, so the source code can be run natively in a Windows 10 environment.

Aside from recoding the SPARK-ID algorithm, the original plan was to convert the SPARK macros as well, so they can be transferred to an online system. Prior to Phase II, the SPARK 16-bit code could not be put on a networked computer and instead could only be run on a Windows XP machine or XP-emulator. (Prior to Phase I of SPARK, there was full scale use of offline Win XP machines to execute SPARK programs. During Phase I, a virtual machine (VM) was configured, particularly a Virtual Win XP Pro Machine on a Win 10 host operating system using the VMware Workstation 15 Player, chosen because the VM could run SPARK scripts at operational speed. The use of the VM also afforded the AFPC/DSYX SPARK Team more latitude to work remotely during the COVID-19 Pandemic, since SPARK programs could be run on a Win 10 laptop with XP-emulators.) Like the SPARK-ID algorithm, the original concept was to code all the macros in R so they can come in from clearance from DMDC to the HRD transfer directory on the Restricted Drive and move over to the SPARK directory on the network. Prior to Phase II, the SPARK data cleaning

and preparation programs could not be on the Air Force Network because they could not be processed there; however, the leftward shift in the project's timeline resulted in formatting all SPARK macros for the AF Network, to include SPARK layouts, utilities, code generation, auditing, ad hoc special purpose, and SPARK-ID generation.

6.0 RECOMMEND TECHNOLOGIES FOR HOSTING/SOFTWARE

As stated previously, the research from SPARK Phase I served as the foundation for Phase II, and the requirement for the AFPC/DSYX's I/O psychologists is to transition to an open-source programming language so they can attain User Group software specific to their field to enhance their collective ability to improve the person-job-match result in career fields for optimal AF mission capability. I/O psychology is an emergent field and the current SAS and SPSS applications will not necessarily respond to the needs of the AFPC/DSYX's professional team, which are often outside the mainstream requirements met by proprietary statistical packages. Moreover, if AFPC/DSYX obtains case- and task- clustering in their toolset, the I/O psychologists would become more innovative contributors to their user group, which would in turn attract the best and most qualified students to enhance the work of the Strategic Research and Assessment Branch. For these reasons, the requirement was to adopt a platform with a standardized code base that supports the current programming in SAS 9.4, as well as open-source solutions like Python and R.

7.0 AFPC/DSYX R CODE REPOSITORY

Building, documenting, and managing the R Code Repository has continued unabated throughout SPARK Phase II, to include Comprehensive R Archive Network (CRAN) Task Views, which is a network of web servers with R source code and manuals, each with an identical copy of all information on CRAN, which is why each web server is called a mirror. In the past, there were thousands of packages on the CRAN; however, now commonly used packages are in curated lists (which are topologies organized by volunteers) and they are called CRAN Task Views. The following CRAN Task Views relevant to AFPC/DSYX's I/O psychologists and for the SPARK Team are included in the R Code Repository: Databases with R, Model Deployment with R, Meta-Analysis, Bayesian Inference, Graphic Displays and Dynamic Graphic Devices and Visualization, Psychometric Models and Methods, Functional Data Analysis, Statistics and Survey Methodology, Probability Distributions, Cluster Analysis and Finite Mixture Models, Design of Experiments and Analysis of Experimental Data, Extreme Value Analysis, Machine Learning & Statistical Learning, High-Performance & Parallel Computing with R, Dire: Linear Regressions with Latent Outcome Variable; Multivariate Statistics; Missing Data (Identification, Manipulation, Evaluation), Numerical Mathematics, Clinical Trial Design, Monitoring, and Analysis, Process/Analyze Tracking Data, Robust Statistical Methods, and gRaphical Models in R.

The CRAN Task Views culled from the topologies organized by R code developers and volunteers were submitted to the SPARK Program Manager and reside on the AFPC/DSYX Restricted Drive.

8.0 SPARK MODERNIZATION EFFORT

Toward the end of SPARK Phase I and the beginning of SPARK Phase II, the AFPC/DSYX SPARK team and the AFPC Data Scientist researched and evaluated possible alternatives to replace the existing SPARK processes with new programming and hosting capabilities that facilitate data importation and integration for AFPC/DSYX data. For any design effort, the focus must first begin with the "as-is" state of the SPARK 1.0 system, which is a legacy system based on flat files and layouts. The SPARK PM has been developing tools throughout the years to manage the data, i.e., to manipulate the layouts. There is a layout for each flat file and when presented with a project, the SPARK PM must discern which flat files to merge together to produce a dataset that can inform the particular research requirement. To be clear, a flat file is normally data written into a file, not indexed or searchable by a database management system. The limitation here, of course, is that flat files are rarely used for a production application, they are not used for scale, so as AFPC/DSYX gets more and more records in the future, the system will become slower as it gets larger. The AFPC/DSYDT data scientist sought solutions to accommodate all of A1, while the SPARK Team focused on AFPC/DSYX requirements (e.g., data analytics framework that can connect to open-source interfaces) and exploring software capabilities to match the requirement. The team explored opportunities for the current SAS 9.4 software system to integrate with open-source solutions by conducting software/platform reviews. Relative to the data and software requirements for all of A1, those for AFPC/DSYX were modest:

- 3 TB for data import clean-room and normalization;
- 120 GB for Master File and Psychometric Data Quick Access Library;
- 900 GB for processing project requirements

Due to the relatively small SPARK footprint, the SPARK PM directed the team to collaborate with the AFPC Data Scientist to pursue AFPC/DSYX requirements within the overall A1 effort and the confines of the AF/A1 data management objective to align any new architectural data structure with the legacy systems within the existing enterprise. Consequently, both the AFPC/DSYX SPARK Team and the AFPC Data Scientist were focused on integrating the current SAS 9.4 application with open-source solutions. The team supported the Data Scientist in analyzing performance factors of different systems, and emphasizing memory as a key feature in A1's server solution, since the amount of memory is directly translatable to the size of the tables that can be handled. The server needed to be large enough to handle the load years in the future and must also have features that encompass the need for expected growth. The team considered many Relational Database Management System (RDBMS) software as well as web-based delivery options as Software as a Service (SaaS). (The SaaS is a method of software delivery that allows data to be accessed from any device with an Internet connection and a web browser. In this web-based model, software vendors host and maintain the servers, the databases, and the code that comprise the application.) Due to Information Assurance requirements that became a significant source of risk for DSYX, the design phase of SPARK II was streamlined in order to focus on the more pressing programming efforts to get the SPARK system on the Air Force network; yet, the following possibilities were the final contenders to modernize the SPARK system as well as the DBMS for all of A1: Oracle Database 19C, Microsoft's SQL Server, Tableau Server, and SAS Viya.

8.1 Oracle Database 19C

The Oracle Database 19C is a completely scalable RDBMS system that allows data objects to be accessed by users using the Structured Query Language (SQL). From the AFPC/DSYX standpoint, the greatest draw to the Oracle Database 19c is the automatic indexing based on SQL tuning methods. The RDBMS identifies candidate indices, verifies their effectiveness, performs and online validation, and implements the indices where appropriate. The tuning activities are auditable with Oracle's reporting capabilities. Furthermore, Oracle's partnership with Microsoft affords the opportunity to deploy Oracle software in Microsoft's public and private clouds while maintaining customer support from Oracle. (The "C" in 19C stands for "Cloud".) Another benefit for the AFPC/DSYX SPARK Team would be to no longer have to manually perform regression tests on new query execution plans prior to upgrading to a newer version of the DBMS, since Oracle 19C automatically checks built-in execution plans against existing ones, and then automatically replaces those that would benefit from the upgrade and retains those that would be slower from the upgrade. Ultimately, however, the Oracle Database 19C was not chosen, since the requirement was for a smaller and less complex system.

8.2 Microsoft SQL Server

Microsoft SQL Server is also a RDBMS system based on the SQL programming language that Database Administrators (DBAs) use to manage the system and query the data they contain. SQL Server is relatively easy to install (via a setup wizard) and is highly secure, both benefits for AFPC/DSYX. Specifically, the RDBMS follows a table structure on rows that connect functions and data elements, which help secure the data within the system. Moreover, SQL Server does not allow any process to access and manipulate the database files at run time and also requires users to perform specific functions or manipulate files by executing an instance, both features enhancing the security of the system. The primary difference between the programming languages used by Oracle 19C and Microsoft SQL Server is that the former uses Procedural Language/SQL (PL/SQL) which can group procedures into packages, while the latter uses Transact-SQL (T-SQL) which is simpler and easier to use. Nonetheless, despite the relative simplicity and ease of use (compared to the Oracle system), Microsoft SQL Server was cost-prohibitive and some considered SQL to have a difficult interface vis-á-vis the database.

8.3 Tableau

Tableau is a platform that is used to create interactive visual analytics in the form of dashboards. The major benefit for using Tableau is that graphs and charts can be made with the tool without the need for any coding expertise. Another advantage is that R code and Python can be used with Tableau to enhance visualizations and to build models. Tableau was not chosen because of the high cost and the need for manual dashboard refreshes to automate reports. Furthermore, one of Tableau's strengths—no programming necessary—was also considered a weakness in a different context: Code generation capability is needed for transferring prototypes into production systems.

8.4 SAS Viya

The SAS Viya is an artificial intelligence (AI) analytic and data management platform that runs on a scalable architecture. As mentioned, the AFPC is currently using SAS 9.4 and the most significant core platform difference is in the manner each platform handles distributed processing: SAS Viya leverages the Cloud Analytic Services (CAS) server, and SAS 9.4 uses SAS® LASR™, SAS® High Performance Analytics (HPA) and SAS® Grid Manager. CAS, the cloud-based runtime environment for data management and analytics in SAS Viya, provides security and also allows the user to share data between sessions and provides fault tolerance (i.e., the system continues to operate despite a failure or malfunction). SAS Viya also offers AI-based text analytics to find patterns in unstructured data and Machine Learning to make predictions from the properties found in that unstructured data, capabilities that are not available with our current SAS 9.4 software package. For AFPC/DSYX, SAS Viya is the most optimal platform for the SPARK system for the following reasons:

- SAS Viya has a standardized code base that supports current programming in SAS 9.4, as well as the following other languages: Python, R, Java, and Lua.
- The upgraded platform has the versatility to accommodate the knowledge and proficiencies of a variety of SPARK support personnel with a diversity of backgrounds, now and in the future.
- SAS is currently forming a partnership with Microsoft, which will enhance capability of analytics in the cloud.
- SAS is the only vendor named a leader in all five Forrester Wave Reports in the following domains:
    - ✓ AI-based text analytics
    - ✓ Digital decisioning
    - ✓ Predictive Analytics (PA) and Machine Learning (ML)

Recall that AFPC/DSYX's main criterion was that the platform has a standardized code base that can support other languages, most notably open-source solutions like Python or R. SAS Viya can import R code seamlessly and AFPC/DSYX I/O psychologists would like to be able to use R so they can share algorithms with other professionals in the field. The AI-based text analytics will allow the SPARK team to find patterns and trends in unstructured data, using Machine Learning through SAS Visual Text Analytics. The Digital Decisioning refers to easy-to-use tools for business experts to author decision logic. Both Predictive Analytics and Machine Learning are used to leverage AI in order to optimize decisions at the customer level. In other words, to tailor the technology to the use, rather than an out-of-the-box generic application. While exploring SAS Viya's capabilities, the SPARK Team attended webinars on ML and AI for the Defense Community and noted a lot of mixed meanings over the layman's use of the terms AI, Predictive Analytics, Data Mining and ML; yet, the expert consensus is to describe ML as being focused on prediction, based on known properties learned from the data, while Data Mining is focused on the discovery of previously unknown properties in the data. In the current lexicon, the aggregation of Data Mining and ML comprises the whole of AI. The SPARK team also gained clarity over the concepts from Coursera courses: PA is a subset of ML, though they are often used synonymously. PA is the

use of predictive models to drive decisions that affect outcomes. The upshot: once AFPC/DSYX can get data onto the SAS Viya platform, their analytic capabilities will improve markedly. The following are descriptions of particular SAS Viya features that will be particularly beneficial to the AFPC/DSYX I/O psychologists and SPARK Team:

- Visual Analytics: Interactive data visualizations identify insights buried in data, revealing patterns and trends that may have been missed.
- Visual Statistics: Provision of a visual drag-and-drop interface for quickly creating descriptive and predictive models on data of any size. This feature also provides an interactive programming interface for those who want to code in SAS or other languages, while taking advantage of powerful SAS statistical modeling and machine learning techniques.
- Visual Text Analytics: Embedded data integration and preparation capabilities to help access, integrate, profile, cleanse, and transform data. Text can be imported directly from out-of-the-box data connectors, to include multiple document formats, relational databases, remote file system data sources, local data file types, social media connectors, and Esri (i.e., geographic information systems (GIS) software). The software also includes self-service data visualization capabilities for exploring and understanding the text data.
- Data Preparation: Provision of a simple, interactive user interface designed for self-service data preparation, so that nontechnical users have flexibility to integrate data from virtually any source and then cleanse and prepare the data for analysis quickly and easily. Data can be loaded in memory so multiple users will share the same view simultaneously. Users' data preparation tasks are fully integrated with downstream reporting and analytics processing, all from the same interface.

The following are the SAS Viya installation and configuration requirements for an experimental deployment, for evaluation purposes only:

- Installation and configuration of SAS Viya will be performed in Amazon Web Service (AWS) or equivalent environment
- Installation will be performed on a single machine; if utilizing AWS infrastructure, SAS recommends the following AWS instance to achieve desired performance:
    - ✓ i3en.12xlarge: 48 vCPUs, 24 cores, 384 GB Random Access Memory (RAM), 4 x 7500 GB NVMe SSD
    - ✓ https://aws.amazon.com/ec2/instance-types/i3en/
- The recommended hardware sizing is based on assessment of analytic functions and users required by AFPC for the Proof of Value (POV)
- User estimates for the POV environment will be approximately 8 total users, which includes 4 "heavy" concurrent users
- Software is licensed as scoped for 12 months at which time, user could choose to discontinue use, renew at current Government Service Administration renewal rate without environment changes, or could choose to move to next phase, which may require rescoping

of # of environments, individual platform sizes, and/or SAS products included in the environment.

Only a few months into the SPARK Phase II project, the AFPC Team eliminated all the other possibilities to include the aforementioned three contenders (Oracle Database 19C, Microsoft SQL Server, and Tableau) and instead decided to leverage the relatively low cost of the SAS Viya platform. Originally, the AFRC offered to pay for half of the SAS Viya purchase to procure a partnership with the AFPC, but eventually A9 and the AETC/SAS joined the partnership as well, further mitigating the cost of the transition from SAS 9.4 to SAS Viya. Aside from the lucrative offer from SAS to customers who already have the SAS 9.4 desktop, the proposal from SAS included unlimited core count and an unlimited number of environments for both SAS Viya and SAS 9.4, which could be used AF-wide. Furthermore, the maintenance cost for sustaining SAS 9.4 each year will also mitigate the final cost for the adoption of the SAS Viya platform. At the time of this writing, the developmental sandbox for the Common Analytical Platform for the Air Force team is anticipated by early September 2021, although the AFPC TMDE requirement for SAS Viya will delay deployment until April 2022, due to the necessity of the FedRAMP IL-4 authorization to operate. The particular SAS product to be purchased is called Evolve Your Analytic Platform (EYAP) and will modernize databases and analytical platforms for both the Air Force and the Space Force.

9.0 SPARK SYSTEM 2.0

Aside from supporting the AFPC data scientist in the SAS Viya procurement (which included building presentations for the SAS Viya proposal for the Senior Advisor/Liaison to the AFPC/DSYX Chief Information Officer (CIO) and the AFPC CIO, facilitating FedRAMP requirements for IL-4 in the phases of implementation, and offering internal and external advisement on the cloud migration process), the SPARK Team had to also focus on the significant security liability for AFPC/DSYX mentioned earlier in this report. Due to the delays in the SAS Viya deployment caused by the FedRAMP requirement, funding issues, and the COVID-19 pandemic, converting the SPARK 1.0 system to R code changed to compiling the SPARK Turbo Pascal macros to Win 10 versions that can be used on the Air Force network even sooner than the original December 2021 suspense date. The initial plan was to rewrite the SPARK-ID algorithm and all the SPARK macros in R code during Phase III, once the new platform was procured, so the SPARK 1.0 System can be put on the Air Force network for improved data security. Yet, rewriting all the SPARK macros to R code to get the programs on the network proved slow and cumbersome; after all, most of the SPARK macros are data manipulations: They are reading in formatted ASCII data files, applying some cleaning and formatting functions, and then writing out the data into a different standardized file format. None of the Turbo Pascal source code is doing any statistical functions or plotting where R code might be required, so to comply with the IA requirements at the soonest, the SPARK Team searched for a freeware Turbo Pascal compiler to convert the legacy code to a format that can run natively in a Windows 10 environment. The team found a Freeware Pascal compiler for Windows 10 that is part of the “Lazarus” software package, and had success compiling the Pascal code. Consequently, each of the SPARK macros (post-2004) were reformatted to Win 10 versions. Build Notes were submitted to AFPC/DSYX for each of the SPARK programs that were compiled, to include those that required modifications in format or logic prior to compilation.

Most of the SPARK macros that would not initially compile were due to the MS DOS specific date function, yet a work-around was used to compile those as well. Rewrites of the portion of each program that would not compile with the non-MS DOS code were necessary, where the GETDATE procedure was utilized. All source code rewrites that had to be made to get a program to compile were included in the Build Notes. (Some of these required replacing the DOS only program calls with alternative code methods that worked on the Win 10 compiler.) Though the focus was on the SPARK macros written after 2004 to include SPARK programs facilitating the use of the SPARK-ID, the team also tested the SPARK PM's CODAP (Computerized Occupational Data Analysis Program) programs with the Win 10 compiler, and those were successful, too. One issue that initially plagued the team were runtime errors, but the Win 10 compiler can also utilize Turbo Pascal Mode, which resolved any runtime errors, provided there were no bugs in the original source code of those particular programs. (The SPARK PM was provided the syntax for the Turbo Pascal mode, so he can use the Lazarus compiler in the future, if necessary.) To use the CHEKLENG.PAS program as an illustration, there were two common types of runtime errors found in the SPARK macros. The first type of runtime error would occur when the input/output files were attempted to be opened, yet they did not exist to be opened. For example, running "CHEKLENG FILE_IN.TXT FILE_OUT.TXT" results in a runtime error if either of the two files do not exist, because no logic exists in the program to test if they exist before operations are attempted on them. Those would be expected runtime errors that can be avoided by giving the right input files to the program. The other type of runtime error (again, to use CHEKLENG.PAS as an example) was that in some cases variables were used in the original Turbo Pascal source code for array indices that were not initialized properly. Adding the proper initialization removed the runtime error. The fact that this error never presented with the original compilations on the XP machines or XP emulators seemed peculiar; however, that perceived oddity was the impetus for discovering that the Lazarus Win 10 compiler has a Turbo Pascal mode that allows more latitude for the inexact coding. If there are more programs in the future that have runtime errors even when given the correct command line attributes, they can be compiled in Turbo Pascal mode to resolve the issue. The Proof of Concept was completed on the remaining SPARK macros prioritized by the SPARK PM; yet, AFPC/DSYX is still in the process of procuring a waiver to put all of the compiled SPARK 2.0 programs on the Air Force Network at the time of this writing. (See Appendix E for the Build Notes.)

10.0 RESULTS AND DISCUSSIONS

This report documents the second and final phase of the Strategic Policy Analysis Resource and Knowledgebase project, which included a recommendation for new programming and hosting capabilities as well as the implementation of the programming recommendation. The SPARK I report (Weissmuller & Cazares, 2020) documented the design philosophies and detailed implementation of the software used in the SPARK system, with the intent of describing the elements and goals that must be preserved in any replacement configuration. Many of these technical approaches go back to Turbo Pascal 2.0 (16-bit software) and use MS-DOS as the bridge between batch files; yet, these approaches are not supported or permitted on the current USAF Network. This means that prior to Phase II, SPARK data processing had to take place on Windows XP PCs (which are outdated and no longer supported by Microsoft) and/or Windows XP-emulators.

The first effort in Phase I of this project was to find an XP-emulator that would run on a modern I7 grade laptop. This stop-gap measure was pursued to avoid mission failure in the event of the actual XP "SPARK" Machine's no longer working and/or being available as an option. After numerous tests with a variety of Virtual Machine (VM) applications, the VMware Workstation 15 Player was chosen, since scripts could be run at operational speed with this particular Virtual Win XP Pro Machine on a Win 10 host operating system. Instructions on running this VM are included in Appendix A.

SPARK Phase II was conceived as a design phase, to assess the future platform for the SPARK programs and other resources. The requirement for the Air Force Personnel Center Strategic Research and Assessment branch is to update their enterprise software from the current SAS 9.4 analytics, reporting, administration, and coding capabilities to a more modern, scalable software platform. The new capability must include the current SAS 9.4 statistical analysis capability as well as enhanced features, such as predictive modeling, text analytics, and ML. Furthermore, AFPC would like to leverage the open-source capability offered by Python, R, Java, etc., in the near future, so an open-architecture is required for data manipulation and analytics development to be callable via the open-source programming language that is ultimately chosen. The open-source programming language capability would allow AFPC's I/O psychologists to share code with other researchers who might suggest improvements to the tools they are currently using. The SPARK team and the AFPC data scientist reviewed several software applications and platforms, and decided that the SAS Viya platform would most optimally fulfill AFPC requirements at a reasonable price. SAS Viya offers AI-based text analytics to find patterns in unstructured data, and Machine Learning to make predictions from the properties found in that unstructured data, capabilities that are not available with AFPC's current SAS 9.4 software package. AFPC joined a partnership with A9, the AFRC and AETC/SAS to procure the modernized Common Analytical Platform. The developmental sandbox for SAS Viya is anticipated by September 2021.

Aside from reviewing and analyzing software requirements and evaluating a multitude of alternatives with criteria from a variety of different command structures in partnership with the AFPC, the AFPC/DSYX SPARK Team also built, documented, and managed the R Code Repository, assisted the AFPC Data Scientist with the FedRAMP requirements for IL-4 authorization to operate as well as the cloud migration process for the new platform, and compiled and/or rewrote all of the SPARK macros (post-2004) so the programs can run natively in a Windows 10 environment and be used on the Air Force Network. Information Assurance requirements necessitated moving SPARK Phase III code production to Phase II, and getting SPARK data on the mainframes became the most critical AFPC/DSYX objective for the SPARK Team. Recoding the legacy SPARK 1.0 programs was essential for getting the system online and for completely eliminating the need for stand-alone machines and for the USBs that were being used to port data from the offline scanner. (Prior to SPARK Phase II, the 16-bit SPARK software that runs on the offline XP machines or XP-emulators were connected to USB drives, which were required to store

the large volume of SPARK data. SPARK files could never be on the Air Force network, because they could not be processed there.) At the conclusion of SPARK Phase II, all the data preparation and SPARK-ID, and SPARK-ID related programs were updated to run in a Windows 10 environment using the Turbo Pascal Mode of the Freeware Pascal compiler, which is part of the Lazarus software package. (Those programs that would not initially compile were rewritten or reformatted to compile.) At the time of this writing, all SPARK system programs can now be run natively in a Win 10 environment on the Air Force Network, and once AFPC/DSYX is able to procure a waiver for the updated macros, the SPARK 2.0 system will be online.

11.0 CONCLUSION

The original SPARK Human Resources Research Databank was developed by the Air Force Human Resources Lab in the 1960s and has evolved over the years. I/O psychology is an emergent field and the current SAS and SPSS applications do not adequately respond to the needs of the AFPC/DSYX's I/O psychologists, which are often outside the mainstream requirements met by proprietary statistical packages. Furthermore, the SPARK 1.0 was a legacy system, based on flat files and layouts. The SPARK PM developed tools throughout the years to manage the data and to manipulate the layouts. There is a layout for each flat file, and when presented with a project, the SPARK PM must discern which flat files to merge together to produce a dataset to inform the particular research requirement. (A flat file is normally data written into a file, not indexed or searchable by a database management system. The limitation here, of course, is that flat files are rarely used for a production application—they are not used for scale—so as the number of records increase, the system becomes slower as it gets bigger.) To accommodate the requirements of both the AFPC/DSYX I/O psychologists and those of the SPARK team, a system-wide update was needed. SPARK 1.0 documented the legacy system and procedures, and SPARK 2.0 analyzed and evaluated the current requirements to recommend a way forward for replacing the existing SPARK processes with new programming and hosting capabilities that facilitate data importation and integration for DSYX data. The DSYX SPARK team joined the AFPC data scientist in matching system requirements with software capabilities and chose the SAS Viya platform in a partnership with A9, AFRC, and AETC/SAS.

Along with the design effort to modernize the SPARK system, it was necessary to shift the project timeline left to begin the programming effort anticipated for Phase III: The offline computers and use of physical media had become a significant source of risk for AFPC/DSYX, as well as the need to protect PII by converting to a SPARK-ID. As with all the other SPARK macros, the programs related to the SPARK-ID were also written in the MS-DOS version of Turbo Pascal programs that can only run on Win XP machines or on host machines with the XP-emulators that were configured during Phase I of SPARK. The original plan was to rewrite the SPARK-ID algorithm and all the SPARK macros in R code, so the SPARK system can be put on the Air Force network for improved data security. Yet, most of the SPARK macros are data manipulations; they are reading in formatted ASCII data files, applying some cleaning and formatting functions, and then writing out the data into a different standardized file format. None of the Turbo Pascal source code is doing any statistical functions or plotting where R code might be required, so to comply with the AFPC/DSYX security requirements, the SPARK Team used a Freeware Pascal compiler for Windows 10, which is part of the Lazarus software package. Using the Turbo Pascal Mode, the team had success compiling the outdated SPARK code to programs that run natively on Windows 10. Build Notes were submitted to the SPARK PM for each program that needed to be modified in format or logic in order to be compiled.

Consequently, instead of the December 2021 suspense for getting the SPARK database fully on the network to fulfill the Information Assurance requirements as stated by AFPC/DSYX branch chief, the SPARK 2.0 system can run on the Air Force Network once AFPC/DSYX procures a waiver. In conclusion, the three-year SPARK effort was completed in only two years

12.0 REFERENCES

Air Force Instruction 33–332. (2020, March 10). Information Management. https://irp.fas.org/doddir/usaf/afi33-332.pdf

DoD Scientific and Technical Information Program [22 Aug 2013]. DoDI 3200.12. Retrieved from https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/320012p.pdf?ver=2019-04-30-073122-220.

Lazarus IDE. (2021). Lazarus: The Professional Free Pascal RAD IDE. https://www.lazarus-ide.org/index.php

OPM GOVT-6. (2020). https://www.osec.doc.gov/opog/privacyact/sorns/GOV-Wide/OPM-GOVT-6-opm-sorn-govt-6-personnel-research-and-test-validation-records.pdf

Personnel Assessment Program [16 May 2019]. AFMAN 36-2664. Retrieved from https://static.e-publishing.af.mil/production/1/af_a1/publication/afman36-2664/afman36-2664.pdf.

Privacy Act of 1974. (1974a). The United States Department of Justice. https://www.justice.gov/opcl/privacy-act-1974

Staley, M. R., & Weissmuller, J. J. (1981). *Interrater reliability: The development of an automated analysis tool*. Brooks Air Force Base, TX, TX: Air Force Human Resources Laboratory, Air Force Systems Command.

System of Records Notice (SORN) Guide. (2010). https://www.opm.gov/information-management/privacy-policy/privacy-references/sornguide.pdf

The Comprehensive R Archive Network. (2021). Institute for Statistics and Mathematics. https://cran.r-project.org/

Turbo Pascal: Version 3.0: Reference manual. (1985). Scotts Valley, Ca., CA: Borland International.

Weissmuller, J. J., Barton, B. B., & Rogers, C. R. (1974). *CODAP: Programmer Notes for the Subroutine Library on the Univac 1108*. Ft. Belvoir, VA: Defense Technical Information Center.

Weissmuller, J. J., Barton, B. B., & Rogers, C. R. (1974). *CODAP: Source Program Listings for the Univac 1108*. Ft. Belvoir, VA: Defense Technical Information Center.

Weissmuller, J. J., Cazares, K. A. (2020). *Strategic Policy Analysis Resource & Knowledgebase (SPARK) Version 1.0 System Documentation.* Randolph AFB, TX: Defense Technical Information Center.

APPENDIX A: Instructions for Virtual (Windows XP Pro) Machine

VMware Workstation 15 Player with Veracrypt for Encrypted Files on GUEST Machine

Currently, AFPC/DSYX is using stand-alone Win XP desktop computers because the software to run the SPARK programs is not approved in the Network. (MS DOS version of Turbo PASCAL only runs on 16-bit machines, so must use an XP machine. Windows 7 has an XP-emulator that can run programs; however, cannot complete a script. Currently, DSYX configured a VirtualWin XP Pro Machine on a Win 10 host operating system using the VMware Workstation 15 Player, which can run scripts at operational speed.) The SPARK program is run in Turbo Pascal 3.0, 16-bit. The Open-Source program called Open Office is used, since Open Office is written in Python and so can use Excel, etc.

Below is the algorithm for this work-around solution, running a Win XP virtual machine on a Windows 10 host machine.

Preliminary Instructions to launching Veracrypt

A. Open native USB drive to find the password for encrypted files.
B. Open folder entitled "Inventory".
C. Click on "Date Modified" to order the dates with most current on top.
D. Open the most current Inventory text (txt) file.
E. Click on Edit and select Find.
F. Type name of program listed on connected native USB drive (e.g., OA).
G. Click "Find Next" until password is located.
H. Keep the Notepad window open (but moved to the side) to later cut & paste the password when launching the Veracrypt program.
I. Close Windows Explorer.
J. Disconnect the USB drive, so the window presents in the virtual machine to connect the drive when you connect the cable to the laptop.

Run the Virtual XP Pro Machine: VMware Workstation 15 Player

1. Double-click on VM Workstation on the host machine.
2. Double-click on the Operating System: Windows XP Professional.
3. Connect the USB drive with the 2.0 USB extension cable.
4. Click on "Connect to a virtual machine" when the "New USB Device Detected" window appears, then click OK.
5. Click OK on the window "Removable Devices" (showing devices that can be connected to the VM).

Launch Veracrypt on Guest Machine

6. Open Veracrypt program on the guest machine.
7. Choose a drive letter (e.g., Drive R).
8. Click Select File button.
9. Browse the USB drive that has the encrypted file.
10. Select and open the file (e.g., S3_x.tc).
11. Click the Mount button.
12. Place a checkmark in the True Crypt Mode box.
13. Select Mount Options.
14. Place a checkmark in the Mount Volume as Removable Medium box.
15. Click OK.
16. Input the password for the encrypted file by cutting from the open Notepad window on your screen and pasting to the Veracrypt box provided.
17. Click Display Password to ensure entered correctly.
18. Click OK.
19. Click Exit.

After the mounting completes, the contents of the encrypted file will show as an unencrypted drive letter on the machine.

20. Click on My Computer.
21. Under Devices with Removable Storage, double-click on the SPARK (USB drive) designated by the letter (e.g., R) mounted in Veracrypt.
22. Run programs.

Power-down VM

23. Double-click on Veracrypt.
24. Click Dismount All.
25. Once drive(s) is (are) dismounted, click Exit.
26. Disconnect SPARK USB drive, if desired.
27. Click Player on the VM and select Exit from dropdown menu.
28. Click Power Off.

NOTE: To access a 300,000-character file that is encrypted, e.g., S3.tc (extension "tc" is for true-crypt). Go to Veracrypt (true-crypt). Go to restricted drive and call it Drive R. Load true crypt as a removable media.

APPENDIX B: Access to AFPC/DSYX's Restricted Drive

Location of new Restricted Drive Master File System

\\tymx-fs-002v\restricted\DS\DS\DSY\DSYX\9 - CAPS\ SPARK-Data-Masters\ APAT\

APPENDIX C: Point of Contacts (PoCs)

External to DSYX and Internal Job Families

ADSS
Lawrence "Larry" Dohm
Principal Software Engineer
Contractor – G2S Corp
HQ AETC/A5TE
DSN 487-3602 | Comm. 210-652-3602

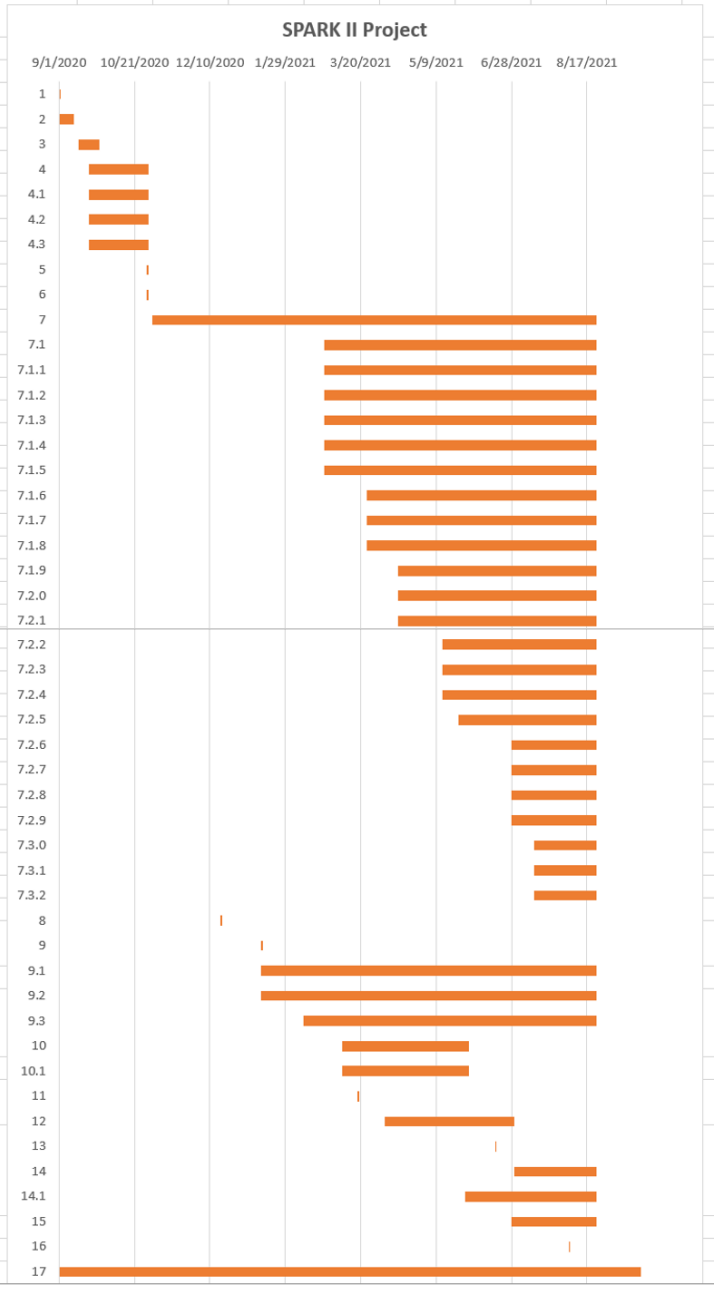
ASVAB
DMDC Reporting System (DMDCRS) Helpdesk
Defense Manpower Data Center (DMDC)
DoD Center Monterey Bay - Seaside, CA
COM +1.831.583.2400 | DSN 312.878.2951
Support Hours: 1000 ET - 1900 ET
https://dmdcrs.dmdc.osd.mil/dmdcrs

(AND)

Tim Powers
Branch Chief
Decision Support/Business Intelligence
Defense Manpower Data Center
(831) 583-2400 x4490

APPENDIX D: SPARK II Gantt Chart

SPARK II Project #	Project Description	Start Date	Duration (Days)	End Date
1	Kickoff Briefing	9/1/2020	1	9/2/2020
2	Documentation Review	8/24/2020	18	9/11/2020
3	SAS 9.4 Software Review	9/14/2020	14	9/28/2020
4	SAS 9.4 System Integration Possibilities	9/21/2020	39	10/30/2020
4.1	Oracle Database 19C	9/21/2020	39	10/30/2020
4.2	Microsoft SQL Server	9/21/2020	39	10/30/2020
4.3	SAS Viya	9/21/2020	39	10/30/2020
5	Recommendation: SAS 9.4 to SAS Viya	10/29/2020	1	10/30/2020
6	Initial Program Management Review	10/29/2020	1	10/30/2020
7	Build, document, & manage R Code Repository for DSYX	11/2/2020	294	8/23/2021
7.1	Comprehensive R Archive Network (CRAN) Task Views	2/24/2021	180	8/23/2021
7.1.1	Databases with R	2/24/2021	180	8/23/2021
7.1.2	Model Deployment with R	2/24/2021	180	8/23/2021
7.1.3	Meta-Analysis	2/24/2021	180	8/23/2021
7.1.4	Bayesian Inference	2/24/2021	180	8/23/2021
7.1.5	Graphic Displays & Dynamic Graphic Devices & Visualization	2/24/2021	180	8/23/2021
7.1.6	Psychometric Models and Methods	3/24/2021	152	8/23/2021
7.1.7	Functional Data Analysis	3/24/2021	152	8/23/2021
7.1.8	Statistics and Survey Methodology	3/24/2021	152	8/23/2021
7.1.9	Probability Distributions	4/14/2021	131	8/23/2021
7.2.0	Cluster Analysis & Finite Mixture Models	4/14/2021	131	8/23/2021
7.2.1	Design of Experiments & Analysis of Experimental Data	4/14/2021	131	8/23/2021
7.2.2	Extreme Value Analysis	5/13/2021	102	8/23/2021
7.2.3	Machine Learning & Statistical Learning	5/13/2021	102	8/23/2021
7.2.4	High-Performance & Parallel Computing with R	5/13/2021	102	8/23/2021
7.2.5	Dire: Linear Regressions with Latent Outcome Variable	5/24/2021	91	8/23/2021
7.2.6	Multivariate Statistics	6/28/2021	56	8/23/2021
7.2.7	Missing Data (Identification, Manipulation, Evaluation)	6/28/2021	56	8/23/2021
7.2.8	Numerical Mathematics	6/28/2021	56	8/23/2021
7.2.9	Clinical Trial Design, Monitoring, & Analysis	6/28/2021	56	8/23/2021
7.3.0	Process/Analyze Tracking Data	7/13/2021	41	8/23/2021
7.3.1	Robust Statistical Methods	7/13/2021	41	8/23/2021
7.3.2	gRaphical Models in R	7/13/2021	41	8/23/2021
8	Quarterly Program Management Review	12/17/2020	1	12/18/2020
9	SAS Viya Procurement Delay/FedRAMP Requirement	1/13/2021	1	1/14/2021
9.1	Review FedRAMP requirements & roadmap	1/13/2021	222	8/23/2021
9.2	Facilitation of FedRAMP Authorization for IL-4	1/13/2021	222	8/23/2021
9.3	Overview of FedRAMP Security Controls for SSP	2/10/2021	194	8/23/2021
10	Conversion of PII to SPARK ID	3/8/2021	84	5/31/2021
10.1	Implement Randomized SPARK ID in lieu of SSAN for 4+ Job Families	3/8/2021	84	5/31/2021
11	Quarterly Program Management Review	3/18/2021	1	3/19/2021
12	Design/Implement network-based capability using R to generate SPARK IDs on criterion datasets that are not included in SPARK Job Families	4/5/2021	86	6/30/2021
13	Quarterly Program Management Review	6/17/2021	1	6/18/2021
14	Eliminate need for physical media currently used as SPARK repository and transfer data to networked computer	6/30/2021	54	8/23/2021
14.1	Compile SPARK Turbo Pascal macros (to include SPARK ID programs) that are currently running on 16-bit software on Win XP machines and XP-emulators run in a Win 10 & Win 12 environment using Lazarus software.	5/28/2021	87	8/23/2021
15	Software as a Service (SaaS) Options with Viya	6/28/2021	56	8/23/2021
16	Final Program Management Review	8/5/2021	1	8/6/2021
17	SPARK II Final Report	8/24/2020	394	9/22/2021



APPENDIX E: Build Notes for SPARK 2.0 Macros

Build Notes for compiling Win 10 programs from the Turbo Pascal macros cited in the SPARK 1.0 Report (Appendix G's SPARK Software Categories & Commands) as well as additional macros submitted for compilation by the SPARK PM during the current SPARK Phase II. Used the Win10 Freeware Pascal Compiler and then modified and/or rewrote SPARK 1.0 macros when necessary. (Compiler available at https://www.lazarus-ide.org/ and freepascal.org.)
All SPARK programs (post-2004) have been compiled to the Win 10 versions for the SPARK 2.0 system.

Utility Programs (used for file processing, usually INPUT/OUTPUT files) (19 programs of this type)

1. **COPYCRLF.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\CopyFile_Stuff
fpc -I..\..\INC -I..\..\REC COPYCRLF.PAS
copy COPYCRLF.exe %Win10Compiles%
```

2. **COPYEQUX.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\CopyFile_Stuff
fpc -I..\..\INC -I..\..\REC COPYEQUX.PAS
copy COPYEQUX.exe %Win10Compiles%
```

3. **COPYFORC.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\CopyFile_Stuff
fpc -I..\..\INC -I..\..\REC COPYFORC.PAS
copy COPYFORC.exe %Win10Compiles%
```

4. **COPYFORK.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\CopyFile_Stuff
fpc -I..\..\INC -I..\..\REC COPYFORK.PAS
copy COPYFORK.exe %Win10Compiles%
```

5. **COPYFOSM.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\CopyFile_Stuff
fpc -I..\..\INC -I..\..\REC COPYFOSM.PAS
copy COPYFOSM.exe %Win10Compiles%
```

6. **COPYGE.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\CopyFile_Stuff
fpc -I..\..\INC -I..\..\REC COPYGE.PAS
```

copy COPYGE.exe %Win10Compiles%

7. COPYMERG.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\CopyFile_Stuff
*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.
fpc -I..\..\INC -I..\..\REC COPYMERG.PAS
copy COPYMERG.exe %Win10Compiles%

8. COPYT2TL.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\CopyFile_Stuff
fpc -I..\..\INC -I..\..\REC COPYT2TL.PAS
copy COPYT2TL.exe %Win10Compiles%

9. CSV2COMP.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC CSV2COMP.PAS
copy CSV2COMP.exe %Win10Compiles%

10. CSV2FIX.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC CSV2FIX.PAS
copy CSV2FIX.exe %Win10Compiles%

11. CSV2FIXC.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC CSV2FIXC.PAS
copy CSV2FIXC.exe %Win10Compiles%

12. CSV2FLDS.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC CSV2FLDS.PAS
copy CSV2FLDS.exe %Win10Compiles%

13. CUTTILDE.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\File-Edits
fpc -I..\..\INC -I..\..\REC CUTTILDE.PAS
copy CUTTILDE.exe %Win10Compiles%

14. FORKSAME.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004

****** The fpc compiler always tries to compile a SYSTEM.PAS file if in current working directory
****** so, because there is a file with that name in SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004 dir
****** it breaks compiling any other programs in that directory. Changed name from SYSTEM.PAS to SYSTEM.PAS.BAK.
fpc -I..\INC -I..\REC FORKSAME.PAS
copy FORKSAME.exe %Win10Compiles%

15. MAT2FIX.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Generic-File-Reformats
*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.
fpc -I..\..\INC -I..\..\REC MAT2FIX.PAS
copy MAT2FIX.exe %Win10Compiles%

16. MAT2FIXC.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Generic-File-Reformats
*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.
fpc -I..\..\INC -I..\..\REC MAT2FIXC.PAS
copy MAT2FIXC.exe %Win10Compiles%

17. MERMATVC.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\MERGING-Matching-Files
fpc -I..\..\INC -I..\..\REC MERMATVC.PAS
copy MERMATVC.exe %Win10Compiles%

18. MIX2PAD.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
fpc -I..\..\INC -I..\..\REC MIX2PAD.PAS
copy MIX2PAD.exe %Win10Compiles%

19. POKEONLY.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004
****** The fpc compiler always tries to compile a SYSTEM.PAS file if in current working directory
****** so, because there is a file with that name in SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004 dir

****** it breaks compiling any other programs in that directory. Changed name from SYSTEM.PAS to SYSTEM.PAS.BAK.
fpc -I..\INC -I..\REC POKEONLY.PAS
copy POKEONLY.exe %Win10Compiles%

Psychometric or Special Purpose (2 programs of this type)

1. **DMDC123B.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004
****** The fpc compiler always tries to compile a SYSTEM.PAS file if in current working directory
****** so, because there is a file with that name in SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004 dir
****** it breaks compiling any other programs in that directory. Changed name from SYSTEM.PAS to SYSTEM.PAS.BAK.
*** Line 753 had no closing "}" on a comment. Edited and ensured the "}" was used so that it compiled.
*** Line 755 had a bug using a "[" where a "{" was called for. Edited and ensured the "{" was used so that it compiled.
*** Line 758 had a bug using a "]" where a "}" was called for. Edited and ensured the "}" was used so that it compiled.
fpc -I..\INC -I..\REC DMDC123B.PAS
copy DMDC123B.exe %Win10Compiles%

2. **FORK4SQL.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004
****** The fpc compiler always tries to compile a SYSTEM.PAS file if in current working directory
****** so, because there is a file with that name in SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004 dir
****** it breaks compiling any other programs in that directory. Changed name from SYSTEM.PAS to SYSTEM.PAS.BAK.
fpc -I..\INC -I..\REC FORK4SQL.PAS
copy FORK4SQL.exe %Win10Compiles%

Layout Generator (.LAY) or Layout-driven Output (.SAS, .SPS, .LAY) (5 programs of this type)

1. **CSV2LAY.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff\Other-Layout-Items

fpc -I..\..\..\INC -I..\..\..\REC CSV2LAY.PAS
copy CSV2LAY.exe %Win10Compiles%

2. LAY2CSV.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.
*** Added " MYINDEX : Integer;" to line #131 , replaced "SC2" with "MYINDEX" on lines #646 and #650.
*** This program has dependencies on MSDOS calls for getting date/time. Created a new GETDATE procedure for this.
*** added "uses sysutils;" to line 38 for new getdate function
*** replaced getdate MSDOS function with the following:
*** PROCEDURE GETDATE;
*** {Create Date/Time. Needs DATETIME as a String in Main Program}
*** var
*** MyTime: TDateTime;
*** BEGIN { GET_DATE }
*** MyTime:= Now;
*** DATETIME := FormatDateTime('dd mmm yy hh:nn', MyTime);
*** END; { GETDATE }
fpc -I..\..\INC -I..\..\REC LAY2CSV.PAS
copy LAY2CSV.exe %Win10Compiles%

3. LAY2SUFC.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** There were unmatched "{" comment brackets in this PAS source file on line #311. Edited and removed the "{" so that it compiled.
*** This program has dependencies on MSDOS calls for getting date/time... getdate procedure was unref'd, so removed.
fpc -I..\..\INC -I..\..\REC LAY2SUFC.PAS
copy LAY2SUFC.exe %Win10Compiles%

4. MASH2LAY.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff\Other-Layout-Items
fpc -I..\..\..\INC -I..\..\..\REC MASH2LAY.PAS
copy MASH2LAY.exe %Win10Compiles%

5. MASHLEN2.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004
****** The fpc compiler always tries to compile a SYSTEM.PAS file if in current working directory
****** so, because there is a file with that name in

SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004 dir
****** it breaks compiling any other programs in that directory. Changed name from SYSTEM.PAS to SYSTEM.PAS.BAK.
fpc -I..\INC -I..\REC MASHLEN2.PAS
copy MASHLEN2.exe %Win10Compiles%

Layout & Script/Code Generation Macros (usually Directory-driven Outputs (.BAT, .PAS) (5 programs of this type)

1. **LAY2PAS2.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** There were unmatched "{" comment brackets in this PAS source file on line #329. Edited and removed the "{" so that it compiled.
*** There were unmatched "}" comment brackets in this PAS source file on line #587. Edited and added the "}" so that it compiled.
*** This program has dependencies on MSDOS calls for getting date/time... created a new GETDATE procedure for this.
*** added "uses sysutils;" to line 3 for new getdate function
*** replaced getdate MSDOS function with the following:
*** PROCEDURE GETDATE;
*** {Create Date/Time. Needs DATETIME as a String in Main Program }
*** var
*** MyTime: TDateTime;
*** BEGIN { GET_DATE }
*** MyTime:= Now;
*** DATETIME := FormatDateTime('dd mmm yy hh:nn', MyTime);
*** END; { GETDATE }
fpc -I..\..\INC -I..\..\REC LAY2PAS2.PAS
copy LAY2PAS2.exe %Win10Compiles%

2. **LAY2SAS.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** This program has dependencies on MSDOS calls for getting date/time... getdate procedure was unref'd, so removed.
*** This program has dependencies on MSDOS calls for clrscr (clear screen). Added "uses Crt;" to start of program for this.
fpc -I..\..\INC -I..\..\REC LAY2SAS.PAS
copy LAY2SAS.exe %Win10Compiles%

3. **LAY2SASD.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** This program has dependencies on MSDOS calls for getting date/time... getdate procedure was unref'd, so removed.

*** This program has dependencies on MSDOS calls for clrscr (clear screen). Added "uses Crt;" to start of program for this.
fpc -I..\..\INC -I..\..\REC LAY2SASD.PAS
copy LAY2SASD.exe %Win10Compiles%

4. **LAY2SASW.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** This program has dependencies on MSDOS calls for getting date/time... getdate procedure was unref'd, so removed.
*** This program has dependencies on MSDOS calls for clrscr (clear screen). Added "uses Crt;" to start of program for this.
fpc -I..\..\INC -I..\..\REC LAY2SASW.PAS
copy LAY2SASW.exe %Win10Compiles%

5. **LAY2SASX.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** This program has dependencies on MSDOS calls for getting date/time... getdate procedure was unref'd, so removed.
*** This program has dependencies on MSDOS calls for clrscr (clear screen). Added "uses Crt;" to start of program for this.
fpc -I..\..\INC -I..\..\REC LAY2SASX.PAS
copy LAY2SASX.exe %Win10Compiles%

Layout and Utility to Audit/Report File Contents (View/Inspect File; Distribute Contents (2 programs of this type))

1. **LAY2FREQ.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** There were unmatched comment brackets in this PAS source file on line #1433. Edited and added the "}" so that it compiled.
fpc -I..\..\INC -I..\..\REC LAY2FREQ.PAS
copy LAY2FREQ.exe %Win10Compiles%

2. **LAYAUDIT.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Layout-Stuff
*** There was unmatched "{" comment brackets in this PAS source file on line #261. Edited and removed the "{" so that it compiled.
*** This program has dependencies on MSDOS calls for getting date/time... getdate procedure was unref'd, so removed.
fpc -I..\..\INC -I..\..\REC LAYAUDIT.PAS
copy LAYAUDIT.exe %Win10Compiles%

Script/Code Generation Macros (5 programs of this type)

1. **DIR3HXP.EXE**

   cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Directory-based-operations
   fpc -I..\..\INC -I..\..\REC DIR3HXP.PAS
   copy DIR3HXP.exe %Win10Compiles%

2. **DIRLIST.EXE**

   cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Directory-based-operations
   *** Line 535 had no closing "}" on a comment. Edited and ensured the "}" was used so that it compiled.
   fpc -I..\..\INC -I..\..\REC DIRLIST.PAS
   copy DIRLIST.exe %Win10Compiles%

3. **EDITBAT.EXE**

   cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004
   ****** The fpc compiler always tries to compile a SYSTEM.PAS file if in current working directory
   ****** so, because there is a file with that name in SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004 dir
   ****** it breaks compiling any other programs in that directory. Changed name from SYSTEM.PAS to SYSTEM.PAS.BAK.
   fpc -I..\INC -I..\REC EDITBAT.PAS
   copy EDITBAT.exe %Win10Compiles%

4. **GEN_EDIT.EXE**

   cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004
   ****** The fpc compiler always tries to compile a SYSTEM.PAS file if in current working directory
   ****** so, because there is a file with that name in SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004 dir
   ****** it breaks compiling any other programs in that directory. Changed name from SYSTEM.PAS to SYSTEM.PAS.BAK.
   *** Line 117 had no closing "}" on a comment. Edited and ensured the "}" was used so that it compiled.
   fpc -I..\INC -I..\REC GEN_EDIT.PAS
   copy GEN_EDIT.exe %Win10Compiles%

5. **GRAB_IT.EXE**

   cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004
   ****** The fpc compiler always tries to compile a SYSTEM.PAS file if in current working directory

****** so, because there is a file with that name in SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004 dir
****** it breaks compiling any other programs in that directory. Renamed this from SYSTEM.PAS to SYSTEM.PAS.BAK.
fpc -I..\INC -I..\REC GRAB_IT.PAS
copy GRAB_IT.exe %Win10Compiles%

Utilities to Audit/Report File Contents (Output = Audit Display) (3 programs of this type)

1. **CHEK1000.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Check-Length-Records
fpc -I..\..\INC -I..\..\REC CHEK1000.PAS
copy CHEK1000.exe %Win10Compiles%

2. **CHEKLENG.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Check-Length-Records
fpc -I..\..\INC -I..\..\REC CHEKLENG.PAS
copy CHEKLENG.exe %Win10Compiles%

3. **FOUR2FRQ.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Frequency_Distributions
*** This program has a call to an unknown function "MemAvail"
*** line #264 changed to: { MemAvail function no longer exists: xMem := MemAvail; }
*** line #265 changed to: xMem := 1000;
fpc -I..\..\INC -I..\..\REC FOUR2FRQ.PAS
copy FOUR2FRQ.exe %Win10Compiles%

The following SPARK programs were compiled and found in the source code tree, but were not listed in Appendix G from the Spark I report:

1. **PAIR2MAT.EXE**

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Generic-File-Reformats
*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.
*** There was a type issue on line #301 where a character variable was being set by a string. The following update had to be made:
*** add array ref to SaveEdit to give out char 0: EntryM.FldEdit := SaveEdit[0]; { "EntryM.FldEdit" is type char, SaveEdit is a 128 char string... need only 1st char }
fpc -I..\..\INC -I..\..\REC PAIR2MAT.PAS
copy PAIR2MAT.exe %Win10Compiles%

2. **CSV2FLAT.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC CSV2FLAT.PAS
copy CSV2FLAT.exe %Win10Compiles%
```

3. **CSV2FLD2.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC CSV2FLD2.PAS
copy CSV2FLD2.exe %Win10Compiles%
```

4. **CSV2LOA.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC CSV2LOA.PAS
copy CSV2LOA.exe %Win10Compiles%
```

5. **FORK2CSV.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC FORK2CSV.PAS
copy FORK2CSV.exe %Win10Compiles%
```

6. **TILD2CSV.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC TILD2CSV.PAS
copy TILD2CSV.exe %Win10Compiles%
```

7. **SPLITCSV.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Work-with-CSV
fpc -I..\..\INC -I..\..\REC SPLITCSV.PAS
copy SPLITCSV.exe %Win10Compiles%
```

8. **EAD2CRS.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-
reformat_Functions
fpc -I..\..\INC -I..\..\REC EAD2CRS.PAS
copy EAD2CRS.exe %Win10Compiles%
```

9. **FIX2FIX.EXE**

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-
reformat_Functions
fpc -I..\..\INC -I..\..\REC FIX2FIX.PAS
copy FIX2FIX.exe %Win10Compiles%
```

10. FORK2TAB.EXE

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
fpc -I..\..\INC -I..\..\REC FORK2TAB.PAS
copy FORK2TAB.exe %Win10Compiles%
```

11. GPS2EXT.EXE

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
fpc -I..\..\INC -I..\..\REC GPS2EXT.PAS
copy GPS2EXT.exe %Win10Compiles%
```

12. IDS2EXT.EXE

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
fpc -I..\..\INC -I..\..\REC IDS2EXT.PAS
copy IDS2EXT.exe %Win10Compiles%
```

13. KEEP2SAS.EXE

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.
*** This program has dependencies on MSDOS calls for getting date/time. Created a new GETDATE procedure for this.
*** added "uses sysutils;" to line 10 for new getdate function
*** replaced getdate MSDOS function with the following:
*** PROCEDURE GETDATE;
*** {Create Date/Time.   Needs DATETIME as a String in Main Program}
*** var
***   MyTime: TDateTime;
*** BEGIN { GET_DATE }
***     MyTime:= Now;
***     DATETIME := FormatDateTime('dd mmm yy  hh:nn', MyTime);
*** END; { GETDATE }
fpc -I..\..\INC -I..\..\REC KEEP2SAS.PAS
copy KEEP2SAS.exe %Win10Compiles%
```

14. LF2CRLF.EXE

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
*** This program has dependencies on MSDOS calls for clrscr (clear screen). Added "uses
```

Crt;" to start of program for this.
fpc -I..\..\INC -I..\..\REC LF2CRLF.PAS
copy LF2CRLF.exe %Win10Compiles%

15. SAS2DICE.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
fpc -I..\..\INC -I..\..\REC SAS2DICE.PAS
copy SAS2DICE.exe %Win10Compiles%

16. SCAN2PAD.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
fpc -I..\..\INC -I..\..\REC SCAN2PAD.PAS
copy SCAN2PAD.exe %Win10Compiles%

17. SSI2OLF.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
*** This program has dependencies on MSDOS calls for clrscr (clear screen). Added "uses Crt;" to start of program for this.
fpc -I..\..\INC -I..\..\REC SSI2OLF.PAS
copy SSI2OLF.exe %Win10Compiles%

18. TAB2FIX.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\Utility-reformat_Functions
*** There were unmatched "{}" comment brackets in the included FLDDATA.REC PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.
fpc -I..\..\INC -I..\..\REC TAB2FIX.PAS
copy TAB2FIX.exe %Win10Compiles%

19. ENCRRESP.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\SPARK-ID-Encryption
fpc -I..\..\INC -I..\..\REC ENCRRESP.PAS
copy ENCRRESP.exe %Win10Compiles%

20. ENCRRESP-171017.EXE

cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\SPARK-ID-Encryption
fpc -I..\..\INC -I..\..\REC ENCRRESP-171017.PAS
copy ENCRRESP-171017.exe %Win10Compiles%

21. ENCRRESP-171017-1240.EXE

```
cd SPARK_Pgms_03242021\SPARK_Source\all_PAS\Since_2004\SPARK-ID-Encryption
fpc -I..\..\INC -I..\..\REC ENCRRESP-171017-1240.PAS
copy ENCRRESP-171017-1240.exe %Win10Compiles%
```

Additional Macros:

1. ADSS_GEN.EXE

*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.

```
fpc  ADSS_GEN.PAS
```

2. CODENORB.EXE

*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.

```
fpc  CODENORB.PAS
```

3. COPYPOKE.EXE

```
fpc  COPYPOKE.PAS
```

4. CopyRand.EXE

```
fpc  CopyRand.PAS
```

5. CopyRanx.EXE

```
fpc  CopyRanx.PAS
```

6. DROPDUPC.EXE

```
fpc  DROPDUPC.PAS
```

7. DROPDUPS.EXE

```
fpc  DROPDUPS.PAS
```

8. FORKRECL.EXE

```
fpc  FORKRECL.PAS
```

9. GEN.EXE

*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.

```
fpc  GEN.PAS
```

10. MAT2LAY.EXE

```
copy LayFlds.INC .
fpc  -I. MAT2LAY.PAS
```

11. FILEMMDD.EXE

*** There were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.

*** This programs had a date function that needed to be rewritten, since the logic in it is slightly different from previous getdate rewrites.

*** Added "uses sysutils; {for updated getdate}" to line 19 for new getdate function.

*** Added " myYRstr, myMOstr, myDAstr, myHRstr, myMINstr, mySECstr : TextLine; {for use in new getdate function}" to line 92 for new getdate function.

*** Added "   MyTime: TDateTime; {for use in new getdate function}" to line 229 for new getdate function.

*** Commented out line 242 "DOS.AX := $2A00;" through line 254 "SEC := DOS.DX SHR 8;"

*** Inserted following lines after line 254 "SEC := DOS.DX SHR 8;}"

```
***   MyTime:= Now;
***   { populate myYRstr, myMOstr, myDAstr, myHRstr, myMINstr, mySECstr then convert to integers for the program }
***   myYRstr := FormatDateTime('yy', MyTime);
***   YR := StrToInt(myYRstr);
***   myMOstr := FormatDateTime('MM', MyTime);
***   MO := StrToInt(myMOstr);
***   myDAstr := FormatDateTime('dd', MyTime);
***   DA := StrToInt(myDAstr);
***   myHRstr := FormatDateTime('hh', MyTime);
***   HR := StrToInt(myHRstr);
***   myMINstr := FormatDateTime('nn', MyTime);
***   MIN := StrToInt(myMINstr);
***   mySECstr := FormatDateTime('ss', MyTime);
***   SEC := StrToInt(mySECstr);
fpc FILEMMDD.PAS
```

12. FILEDATE.EXE

*** This program had a date function that needed to be rewritten, since the logic in it is slightly different from previous getdate rewrites.

*** Added "uses sysutils; {for updated getdate}" to line 12 for new getdate function.

*** Added " myYRstr, myMOstr, myDAstr, myHRstr, myMINstr, mySECstr : TextLine; {for use in new getdate function}" to line 85 for new getdate function.

*** Added "   MyTime: TDateTime; {for use in new getdate function}" to line 222 for new getdate function.

*** Commented out line 235 "DOS.AX := $2A00;" through line 247 "SEC := DOS.DX SHR 8;"

*** Inserted following lines after line 247 "SEC := DOS.DX SHR 8;}"
*** MyTime:= Now;
*** { populate myYRstr, myMOstr, myDAstr, myHRstr, myMINstr, mySECstr then convert to integers for the program }
*** myYRstr := FormatDateTime('yy', MyTime);
*** YR := StrToInt(myYRstr);
*** myMOstr := FormatDateTime('MM', MyTime);
*** MO := StrToInt(myMOstr);
*** myDAstr := FormatDateTime('dd', MyTime);
*** DA := StrToInt(myDAstr);
*** myHRstr := FormatDateTime('hh', MyTime);
*** HR := StrToInt(myHRstr);
*** myMINstr := FormatDateTime('nn', MyTime);
*** MIN := StrToInt(myMINstr);
*** mySECstr := FormatDateTime('ss', MyTime);
*** SEC := StrToInt(mySECstr);
fpc FILEDATE.PAS

SPARK-ID Related:

1. **FIX2FRQ.EXE**

cd extras6.8.2021
*** this program has a call to an unknown function "MemAvail"
*** line #246 changed to: { MemAvail function no longer exists: xMem := MemAvail; }
*** line #247 changed to: xMem := 65000;
fpc -I..\..\INC -I..\..\REC FIX2FRQ.PAS
copy FIX2FRQ.exe %Win10Compiles%

2. **LAY2CSV.EXE**

cd extras6.8.2021
*** there were unmatched "{}" comment brackets in this PAS source file. Edited and ensured each comment "{" had a "}" so that it compiled.
*** added " MYINDEX : Integer;" to line #137 , replaced "SC2" with "MYINDEX" on lines #646 and #650
*** this program has dependencies on MSDOS calls for getting date/time... created a new GETDATE procedure for this
*** added "uses sysutils;" to line 40 for new getdate function
*** replaced getdate MSDOS function with the following:
*** PROCEDURE GETDATE;
*** {Create Date/Time. Needs DATETIME as a String in Main Program }
*** var
*** MyTime: TDateTime;
*** BEGIN { GET_DATE }
*** MyTime:= Now;
*** DATETIME := FormatDateTime('dd mmm yy hh:nn', MyTime);

*** END; { GETDATE }
fpc -I..\..\INC -I..\..\REC LAY2CSV.PAS
copy LAY2CSV.exe %Win10Compiles%

3. LAY2PII.EXE

cd extras6.8.2021
*** there were unmatched "{}" comment brackets in this PAS source file on line 1425. Edited and ensured each comment "{" had a "}" so that it compiled.
fpc -I..\..\INC -I..\..\REC LAY2CSV.PAS
copy LAY2CSV.exe %Win10Compiles%

4. LAY2TEMP.EXE

cd extras6.8.2021
*** removed getdate MSDOS procedure on line #121 since it is never called in this program
*** there were unmatched "{}" comment brackets in this PAS source file on line 127. Edited and ensured each comment "{" had a "}" so that it compiled.
fpc -I..\..\INC -I..\..\REC LAY2CSV.PAS
copy LAY2CSV.exe %Win10Compiles%

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AAE—Airman Active Extract (Enlisted Personnel File)

ADSS—Automated Data Subsystem

AETC—Air Education and Training Command

AETC/SAS— Air Education & Training Command/Studies /Studies & Analysis Squadron

AF/A1PT—Air Staff Directorate Force Management Policy for Accession & Training

AFCD—Air Force Classification Directory

AAE—Active Airmen Extract (Enlisted Personnel File)

AFCAPS—Air Force Center for Applied Personnel Studies

AFECD—Air Force Enlisted Classification Directory

AFHRL—Air Force Human Resources Laboratory

AFI—Air Force Instruction

AFIT—Air Force Institute of Technology

AFMAN—Air Force Manual

AFOCD—Air Force Officer Classification Directory

AFOMS—Air Force Occupational Measurement Squadron

AFOQT—Air Force Officer Qualifying Test

AFPC—Air Force Personnel Center

AFPT—Air Force Personnel Test

AFRC—Air Force Reserve Command

AFRL—Air Force Research Laboratory

AFRS—Air Force Recruiting Service

AFSC—Air Force Specialty Code

AFSN—Armed Forces Serial Number

AF-WIN—Air Force Work Interest Navigator

APAT—Applied Performance Assessment and Testing (Facility at Lackland AFB)

ASCII—American Standard Code for Information Interchange

ASVAB—Armed Services Vocational Aptitude Battery

AWS – Amaxon Web Service

BAE—Officer Active Extract (Officer Personnel File)

BMT—Basic Military Training

CAS—Cloud Analytic Services

CFM—Career Field Manager

CODAP – Computerized Occupational Data Analysis Program

CoP—Community of Practice

CSV—Comma Separated Value (May substitute a TAB, $, or ^ as the delimiter vice the comma)

CT—Cyber Test

CTM—Controlled Test Material

DBA—Database Administrator

DICE—Data Item Code Explanation

DMDC—Defense Manpower Data Center

DUA—Data Use Agreement

DoD—Department of Defense

DoDI—Department of Defense Instruction

DoD ID—Department of Defense Identifier (cited on back of Common Access Card (CAC))

DRILS—Data Retrieval and Information Locator System

DSYX—Strategic Research & Assessment Branch (HQ AFPC/DSYX)

DTIC—Defense Technical Information Center

EAP—Examining Activities Program (USAF EAP is currently located in HQ AF/A1PT)

EDIPI—Electronic Data Interchange Personal Identifier (now called DoDID)

EMAF—Enlisted Master Applicant File

EYAP—Evolve Your Analytic Platform

FAA—Federal Aviation Administration

FAIR—Findable, Accessible, Interoperable, and Re-usable

FedRAMP-Federal Risk and Authorization Management Program

FIDO—File Item Data Overview

FOUO—For Official Use Only

GDIT—Generalized Data Index Table

GIS—Geographic Information Systems

GTIMS—Graduate Training Integration Management System

HPA—High Performance Analytics

HRRD—Human Resources Research Databank

IL-4-Impact Level 4

I/O—Industrial/Organizational

JCA—Job Compatibility Assessment

MEPS—Military Entrance Processing Station

MilPDS—Military Personnel Data System

MOU—Memorandum of Understanding

MXN—Maintenance Next

OA—Occupational Analysis

OTA—Oracle Training Application

OTS—Officer Training School

PAF—Project Air Force

PCSM—Pilot Candidate Selection Method

PDS—Personnel Data System

PII—Personally Identifiable Information

POE—Point of Entry

PSM—Predictive Success Model

R&D—Research and Development

RAM – Random Access Memory

RDB—Recruit Database

RDBMS—Relational Database Management System

ROTC—Reserve Officer Trainer Corps

RPR—Request for Personnel Research

SAPR—Sexual Assault Prevention Reporting

SaaS—Software as a Service

SAS—Statistical Analysis Software

SIMF—SPARK Identify Master File

SKT—Specialty Knowledge Test

SOR—System of Record

SORN—System of Record Notice

SPAP—Strategic Personnel Assessment Program

SPARK—Strategic Policy Analysis Resource & Knowledgebase

SPARK-ID—Strategic Policy Analysis Resource & Knowledgebase Identifier

SPSS—Statistical Package for the Social Sciences

SQL—Structured Query Language

SSAN—Social Security Account Number

STIP—Scientific and Technical Information Program

TAPAS—Tailored Adaptive Personality Assessment System

TBAS—Test of Basic Aviation Skills

TMDE-Talent Management Data Environment

TUA—Test User Agreement

TIMS—Training Integration Management System

TTMS—Technical Training Management System

TXT—Text (File Extension)

UAR—Uniform Airman Record (Now use AAE)

UMD—Unit Manpower Document

UOR—Uniform Officer Record (Now use BAE)

VM—Virtual Machine

WAPS—Weighted Airman Promotion System

XLS or XLSX—Excel Spreadsheet