FS_DE new test ok.xlsx

1 Introduction

The Data Engineering (DE) comprises all tools related to the provision of engineering and parameter data to the Spectrum Power system (during commissioning and subsequent modifications/extensions). The main modules of Data Engineering are the Spectrum Power GIS Data Import Management (GDIM) and the Information Model Management (IMM) component.

2 GIS Data Import Management

2.1 Functional Overview

GIS Data Import Management (GDIM) enables a Geographic Information System (GIS) to be a source—or the source master—for some data in a Spectrum Power™ 7 (SP7) system._x000D_ GIS Data Import Management (GDIM) transforms and imports maps and engineering data created in a GIS environment into the Information Model Manager (IMM), the SP7 Engineering System. It is a part of a chain that supports the activation in Spectrum Power runtime databases that support Distribution Management System (DMS) functions._x000D_ GDIM supports a GIS import with an IMM-maintained High Voltage (HV) network model for a combined sub-transmission or distribution network. _x000D_ GDIM supports the following modes of import:_x000D_ * GIS initial import (bulk import)_x000D_ This import mode performs a transfer of GDIM data to IMM. All existing data that was imported previously is overwritten. Previously imported data into IMM that no longer exists in GDIM will remain in IMM._x000D_ * GIS incremental import_x000D_ GDIM supports the self-contained incremental import. It determines the differences between two complete GIS data sets and updates the SP7 database accordingly by the addition, deletion or modification of objects._x000D_ * GIS delta import_x000D_ GDIM also supports the external determined incremental import. The request and reception of files describe the increments._x000D_ * GDIM IMM change detection_x000D_ IMM change detection checks for consistency between domain-objects in GDIM and IMM.

2.1.1 Functional Blocks of GDIM

(image: 1-img83a51b366f6fb0799da352393a2de949_1_en_US_TIFF.jpg) _x000D_ Figure 2-# GDIM Functional Blocks Overview_x000D_ The major functional blocks of the GDIM include_x000D_ * Data extraction_x000D_ The data extraction block interfaces with the GIS database, extracts the data, and uses the information to populate the extracted dataset. It also includes execution of validation rules to check the accuracy of the imported data before processing them further in the transformation module. The validation rules include attribute checks (null, range, and enumeration checks) and consistency checks (phase and voltage check, and graphic consistency check). The validation results or findings are displayed in the GDIM User Interface

(UI)._x000D_ For the data extraction from the delivered GIS data, the data structure must be provided in the structure required by GDIM from the GIS system._x000D_ * GDIM UI_x000D_ This comprises functions for management and control of the GDIM workflow, for example, start stop of import, mode of import, enable or disable data validation, and so on._x000D_ * Workflow management_x000D_ This comprises the business logic for the GDIM UI, that is, getting the next command, checking whether undo of the previous step is allowed._x000D_ * Version control_x000D_ Data imported from GIS is maintained in versions. For each new import, a new version is created._x000D_ * Change management_x000D_ For the self-contained incremental import, the change management functionality performs the detection of differences between two versions of extracted GIS data, the previously imported dataset and the new one received from the GIS._x000D_ * Data pre-transformation and transformation_x000D_ This group of functions handles the transformation of the extracted data to the format for the engineering IMM. The transformed data is stored in the extracted dataset. Validations are done based on the transformed data, for example, consistency checking of voltage levels and phase information._x000D_ * Target model output_x000D_ The output writer uses the transformed data and generates XDF files for import into IMM. It is driven by a mapping file which maps the transformed data into the target model._x000D_ * Extracted dataset_x000D_ All data imported from GIS is stored in GDIM in the extracted dataset. The object model of the extracted dataset (consisting of domain, connectivity, and graphics data) represents a customer independent model of the GIS equipment data. The extracted dataset model remains fixed over time, although there may be some occasional model enhancements._x000D_ The following two sets of data are distinguished:_x000D_ * Extracted data_x000D_ * Transformed data_x000D_ Version control is applied to both data sets; a minimum of two versions are available. The difference between these two versions describes the increment or data changes. Only this increment is being forwarded to the engineering system.

2.1.2 Workflow Overview of GDIM

* GIS -> GDIM - The data extraction imports the data from GIS and writes it into GDIM’s extracted dataset. The GDIM extracted dataset is a standardized intermediate schema._x000D_ * GDIM - During extraction and transformation process, validation rules (attribute and consistency checks) are executed to check the accuracy of the imported data._x000D_ * GDIM - Change management detects the changes to the previous version in the extracted dataset._x000D_ * GDIM - Data model transformation transforms the data from the extracted data model as required and appropriate for the engineering system that supplies the Spectrum Power system and stores it in the extracted dataset in the transformed data model._x000D_ * GDIM - The target model output builds XDF files for graphic and domain data._x000D_ * GDIM - The generated XDF files are imported into the engineering IMM._x000D_ * IMM and IMM -> SP7Prepare/Transfer/Activation to Spectrum Power system. GDIM triggers the activation process. This is handled by the workflow controller using the GDIM UI.

2.1.3 GDIM User Interface and Workflow

The GDIM UI visualizes the current progress and shows log/error messages._x000D_ On the GDIM UI, the data engineer selects the mode of operation (bulk, incremental mode or delta mode) and choose between stepwise or automatic import._x000D_ Depending on the settings in the GDIM UI, the workflow coordinates the functions GIS data extraction, change management, data validation, model transformation, and the target model output._x000D_ After the generation of the eXtensible Data Format (XDF) files by the target model output, the workflow controller creates an IMM job and imports the XDF files into this job. The workflow then calls the validation (can be selected optionally), preparation, transfer, activation to the realtime system, and finalization of this job. The user can select whether the validation is called and if the job should be finalized._x000D_ After finalization, the GDIM import is finished and the changes become permanent._x000D_ To start another GIS extraction, the job must be finalized.

2.2 GIS Data Extraction

The focus of the data extraction and import is to extract the GIS data from the GIS source systems and use the retrieved data to create the relevant instances in the extracted dataset, which contains a GIS vendor independent representation of the GIS data as required by GDIM. _x000D_ (image: 1-img6a659f53f93df26f9da3522a36faf07c_2_en_US_TIFF.jpg) _x000D_ Figure # Functional Blocks of the GIS Data Extraction_x000D_ The GIS data extraction offers a modular architecture consisting of:_x000D_ * Translation engine_x000D_ * Extraction mapping driving the extraction process in the translation engine_x000D_ By having these three parts within the extraction module, the tasks within are separated into GIS specific and general tasks. The GIS data must be provided by the customer according to our interface specification. The general part of mapping data from source to destination and updating the objects to the target dataset tasks are part of the translation engine. The source GIS are transparent at this point in the process._x000D_ The interface between the GIS data and the translation engine module is standardized._x000D_ The extracted GIS data in the extracted dataset is stored using predefined Oracle tables. When writing objects to the extracted dataset, the translation engine performs formal data validation rules against the extracted data._x000D_ The extracted dataset contains the data which was extracted from the GIS source systems.

2.2.1 GIS Data Sources

The Data Engineering (DE) comprises all tools related to the provision of engineering and parameter data to the Spectrum Power system (during commissioning and subsequent modifications/extensions). The main modules of Data Engineering are the Spectrum Power GIS Data Import Management (GDIM) and the Information Model Management (IMM) component.

A GIS can utilize multiple sources of data. Each of these sources may need to be considered by GDIM. To absorb the GIS-centric datatypes and formats, the various GDIM is designed to contain special processing and functionality specific to the relevant GIS and data model. The translation engine drives the relevant data into the extracted data set. _x000D_ The following are the scenarios for GIS data sources:_x000D_ * Single source system_x000D_ Data

for all the types are sourced from a single source GIS or format (GIS)._x000D_ * Single source system, static data_x000D_ Only static geometric data and symbology definitions are provided._x000D_ * Geometric data is provided with a different granularity from the base object, electrical, and connectivity data (for example, multiple DXF are provided)._x000D_ Granularity of graphic data might be on:_x000D_ * Entire network level_x000D_ * Voltage level_x000D_ * Substation level_x000D_ * Feeder level, and so on._x000D_ * All data must come from one GIS source system.

2.2.2 GIS Translation Engine

The translation engine governs the extraction process. The translation engine is a generic mapping engine which translates the data types (for objects, attributes, and values) of the GIS of concern to the data types of the extracted dataset. This generic mapping engine is driven by configurable mapping files. During its processing, this engine uses the GIS data and translates it to the corresponding extracted dataset data using the specified extraction mapping. It then writes this mapped data to the extracted dataset._x000D_ The translation engine manages the interaction between the various extraction components. It functions as an interface between the GDIM extracted dataset and the GIS. It receives the data from the provided GIS dataset, and translates it according to the extraction mapping to the intermediate data model of the extracted dataset. _x000D_ Based on directives in the mapping file, the engine does the following: _x000D_ * Translates the GIS data to the corresponding extracted dataset data using the type-translation rules configured in the mapping files_x000D_ * Writes the processed data to the extracted dataset_x000D_ The extracted dataset is the same regardless of the data model in the GIS.

2.2.3 DXF Support

GDIM supports the import of static graphics from a DXF file. The supported DXF file format version is AC1024._x000D_ DXF parser only supports some dxf entities such as LWPOLYLINE, LINE, TEXT, MTEXT, CIRCLE, ARC and there is a limitation on IMM/ODB side according the maximum pointcount of a polyline.

2.3 Extracted Dataset

The extracted dataset, which is an Oracle database, serves as an intermediate repository of all data that has been extracted from the GIS source systems. If multiple source systems are applicable on a project, then the extracted dataset represents the combination of those sources. Multiple source systems are not supported from standard system. The schema of the extracted dataset is a hybrid between a GIS type of database and the engineering IMM database Common Information Model (CIM). It represents a generic superset that is intended to be compatible with any GIS source system that requires to be considered. This is to ensure that detection of incremental updates is as straightforward as possible._x000D_ The extracted dataset may only represent a subset of the original GIS database, especially if filters are applied. _x000D_ For example, non-electrical entities which are not relevant or useful for DMS

purposes may be maintained in the GIS but not extracted to GDIM._x000D_ The extracted dataset is used for storing the GIS data in the GDIM environment for change detection and derivation of the subsequent processing. It holds the current state of the GIS objects or network as of the last extraction. If the data extracted from the GIS was not subject to input filters, then the extracted dataset represents a one-to-one match with the GIS (although the object type or attribute may have changed as a result of mapping between the GIS source system and extracted dataset models). Even though it functions as an intermediate data repository, the extracted dataset is actually another dataset modeled in IMM._x000D_ The main information stored in the extracted dataset includes the following: _x000D_ * Domain and connectivity object data stored in tables resembling those used in the CIM-based engineering IMM dataset_x000D_ * Geometric data stored in each table_x000D_ * Additional graphic-related attributes

2.4 Data Validation

GDIM validates the extracted and re-modeled data. Thus, two separate validations are done in GDIM. Field validation performs object individual checks by inspecting the attributes of the data extracted from GIS._x000D_ Attribute checks: _x000D_ * Null check_x000D_ * Range check_x000D_ * Enumeration check_x000D_ Thus, the quality of the received data is verified._x000D_ After re-modeling the network by transformation, the resulting network is checked for consistency._x000D_ Consistency checks: _x000D_ * Check phase and voltage of connected objects_x000D_ Starting point for phase and voltage tracing should be available, for example, feederhead switch.The network needs to be Feederbased. Connected objects must be in same B1 for this validation to work in standard config._x000D_ * Detection of de-energized equipment_x000D_ * Graphic consistency check

2.5 Change Management

GIS provides bulk data or GIS data describing the increments. When GIS provides bulk data then this requires the GDIM application to detect the relevant changes, which is called self-contained incremental (also called incremental bulk) GIS import. GDIM has to perform change detection (which includes connectivity, attribute, and graphic-related modifications) in order to detect the differences between two versions of the GIS data. This change detection is performed by the change management functionality. The GIS provides the whole set of data, change detection finds the changes and provides only the delta data to further processing and further on to IMM._x000D_ Change detection is supported by version control which is responsible to maintain multiple versions of the GIS data in the extracted dataset._x000D_ For the delta detection, a minimum of two versions of the extracted GIS data are kept in the extracted dataset: _x000D_ * The last version of the GIS data, called version N. This data corresponds to the network data activated in the operational database ODB._x000D_ * The current version of the GIS data, called version N+1. This is the data to be transformed and imported into the engineering system._x000D_ Change management detects the changes between version N and version N+1, which is the delta data between the two versions. _x000D_ The following functions started by the workflow data model transformation processes delta data only._x000D_ This

detection capability allows subsequent steps to only process a changed set of data, which drastically reduces processing time and speeds up the complete process. _x000D_ For GIS delta mode (provision of increments by GIS system), the change management is bypassed, however the basic functions of it are used for creating a new GDIM version.

2.6 Data Transformation

The model transformation functional block processes the delta data of the current version stored in the extracted dataset and performs the necessary transformations, adjustments, and enhancements to the model. The model transformation then generates the corresponding domain and graphics data and stores it as transformed data. The target model output uses the transformed data to generate XDF files for the import into the engineering IMM dataset. _x000D_ (image: 1-imga7321897c0b021269da3522a72a3fc02_1_en_US_TIFF.jpg) _x000D_ Figure # Detailed View of the Model Transformation Functional Block_x000D_ The model transformation transforms the delta data from the extracted dataset into a model suitable for import into IMM. The data is organized in a hierarchically (for example, the substation has a child voltage level, the voltage level has a child equipment, and so on) and relevant operational names are assigned. The hierarchy and names are then used to construct the Technological Addresses (TA) during the propagation of data from IMM to Spectrum Power. Expansion, contraction, and adjustment of the network model are based on configurable rules._x000D_ Graphic objects are assigned to network diagrams and their coordinates are adjusted from geographical coordinates to Spectrum Power network diagram coordinates._x000D_ The data model transformation is of two types, full and incremental. A full (bulk) transformation is required for initial population and the incremental transformation is utilized in everyday business. The full data transformation is done within one version N while for incremental mode the delta between version N and version N+1 is being the input for the model transformation._x000D_ The data model transformation step consists of executing several functions to convert the model schema from what resides in the extracted dataset to what is required in the engineering IMM dataset. It also includes performing several functions to expand or contract the model as required meeting Advanced Distribution Management System (ADMS) operational requirements. _x000D_ The model transformation is functionally and logically split in the following parts: _x000D_ * Pre-transformation_x000D_ Pre-transformation is performed for domain data only. It transforms an edge based into a node based topological model. It consists of topological expansions that are specific to a node-based representation._x000D_ * Transformation_x000D_ Transformation is performed for domain and graphic data. It consists of model expansions, contractions, and adjustments that are transparent from the eventual target system._x000D_ * Graphic adjustment_x000D_ The simple and complex functions classified as part of graphic adjustment are those that do not have an impact on the domain and connectivity data, where the target model or system is transparent._x000D_ The following are examples of complex functions that are performed during the graphic adjustment stage._x000D_ * Layer assignment_x000D_ * Adjustment and shifting of coordinates_x000D_ * Assignment of graphic relationships_x000D_ * Calculation and assignment of attributes and associations_x000D_ * Transformation of coordinates from absolute to relative_x000D_ * Graphic type assignment_x000D_ * Handling and processing

of static and dynamic colors_x000D_ * Creation of annotation_x000D_ * Target model processing_x000D_ Target model processing is performed for domain and graphic data. It consists of model expansions and adjustments that are specific to the target data model, the engineering IMM in this case._x000D_ In addition, the target model processing function is split into two separate stages:_x000D_ * Model adjustment consists of generating additional target-related attributes, associations, and properties._x000D_ For example, TAs, block types, and so on._x000D_ * Model enhancement consists of adding supplementary objects._x000D_ For example, tags, analogs, and so on.

2.7 Configuration

Two central concepts that pertain to the overall GDIM design are as follows: _x000D_ * Configuration-driven_x000D_ * Well-defined, extensible interfaces_x000D_ The GDIM is designed to allow for the core functionality to remain fixed over time. By making GDIM highly configurable, differences in customer data modes can be addressed by changing or expanding the defined configuration files. These configuration files are then used by the generic mapping engine (relevant both for the extraction module and the transformation module) to map from the source model to the target._x000D_ By constructing well-defined, extensible interfaces, additional functions can be developed in the future without impacting existing functionality. _x000D_ Several of the GDIM modules and/or phases are configuration driven. They include the following:_x000D_ * The data extraction_x000D_ * The model transformation_x000D_ * The target model output

2.8 GDIM – IMM Change Detection

The GDIM – IMM change detection checks the consistency between the GDIM and IMM data bases (Spectrum Power 7 engineering system). The last imported GIS dataset is compared with the export from IMM. The result of the compare is a list of equipment that is missing at either side._x000D_ _x000D_ (image: 1-Tip.png) | NOTE The GDIM – IMM change detection checks only the equipment that have a GISTAG. Only equipment imported from GIS to IMM have a GISTAG. Manually added objects in IMM do not have a GISTAG. In the compare only the mutual existence of the objects in GDIM and IMM is checked. Objects attributes, or graphics are not compared.

2.9 Quality Assurance Server

For a controlled system environment, GDIM can run on a Quality Assurance Server (QAS). This alters the workflow, to provide the option to check the imported data on a non-production environment. For the use of this option, a QAS server must be available, and all data that is in the production system must also be present on the QAS IMM (This includes any data that is not imported from GDIM)._x000D_ Workflow_x000D_ After the initial bulk import, GDIM can run from the QAS system to control any GIS data changes provided by GDIM. When a IMM Job is created, it halts the process to enable the user to check if the content of the job is correct and viable for the production system._x000D_ After the user verification is done on the QAS system, the job

can be propagated from the QAS IMM to the production system by the user. After these steps are finished, GDIM takes over, continues the rest of the workflow necessary and continues to prepare the next job if it is configured to split jobs. The splitting of jobs is relevant for large networks, where a high number of changes are delivered with an incremental or delta dataset which exceeds a total number of >15.000 changed instances.

2.10 Database Location and Backup

The GDIM Database and Versioning is in the DOR oracle database, under GDIMU, GDIMU_X and GISU Schemas. Moving the database or producing a backup has to be done by a system administrator and is not automated. In case of a database failure, the version history is lost and has to be restored by a backup. It is not possible to repopulate the GDIM database from IMM._x000D_ If the database is not backed up, GDIM can produce a new version by running a bulk import. Due to processing limitations, the new GDIM version has to be verified against the current IMM by the GDIM – IMM change detection and the GIS data must be adapted to ensure coherency between the naming of objects in the IMM and GDIM database._x000D_ There exists no database backup or mirroring from the Productive server and QAS Server. If GDIM is running on either of these systems, and the database is compromised by hardware or software failure, there is no option to restore it from the other system.

2.11 Workflows

2.11.1 GIS Initial Import

The GDIM system, engineering IMM, and the operational Spectrum Power system is initialized with a bulk export from GIS. This is considered a one-time data migration exercise is done once before the GIS extract workflow is initiated. This process is started on the GDIM UI using the bulk mode. For the initial import GIS extraction and data model transformation is performed, XDF files are written. The further steps to import the data into IMM are done using the GDIM UI.

2.11.2 GIS Incremental Import

* Incremental import supports auto-detection import on the full network model. Data is imported in GDIM in a new extracted dataset. During import, the change management functionality is used for comparing the imported data with the previous version of the data._x000D_ * The identified changes are transformed within the model transformation and XDF files are written._x000D_ * The GDIM UI is used for the import of the XDF files to an IMM job as well as calling the preparation, transfer, and activation to the system._x000D_ _x000D_ (image: 1-Tip.png) | NOTEProper distribution of the changes over all Spectrum Power servers is the responsibility of the Spectrum Power IMM engineering activation, which includes Multisite too.

2.11.3 GIS Delta Import

* In this case, GIS data describing the increments (deltas) are received. GDIM does not do any change detection._x000D_ * The identified changes are transformed within the model transformation and XDF files are written._x000D_ * The GDIM UI is used for the import of XDF files to an IMM job as well as calling for the preparation, transfer, and activation of the system.

2.11.4 GDIM – IMM Change Detection

The GDIM IMM change detection checks the consistency between the IMM and GDIM data bases. The result of the compare is displayed in the GDIM UI.

2.12 Non-Functional Topics

2.12.1 User Interface

GDIM provides a UI that drives and controls the Import process from the GIS and visualizes of the current process status (which phase it is in). The UI can be opened on the ADM server._x000D_ Through the UI the user can select the mode of operation, whether it is a bulk import or incremental import. Also, the selection of either continuous execution or manual stepwise execution is possible. _x000D_ The GDIM UI provides the following features:_x000D_ * Selection of the operation mode: Bulk or incremental import_x000D_ * Processing of static background data using the DXF plug-in_x000D_ * Selection of continuous execution or stepwise execution_x000D_ * Visualization of the state of the import progress_x000D_ * Selection of validation log_x000D_ * Controlling the GDIM workflow including extraction, change management and transformation_x000D_ * Controlling import to IMM, preparation, transfer and activation to the runtime database_x000D_ * Undo and re-do of versions, discard or invalidate versions if data issues are found, and so on_x000D_ * GDIM – IMM change detection_x000D_ (image: 1-img8bb3cc9999af28aa9da3522a670c1d16_1_en_US_PNG.png) _x000D_ Figure 2-# Sample Display - GDIM User Interface

3 Information Model Management

3.1 Functional Overview

3.1.1 Purpose

3.1.2 Model Merge Framework

In power companies, several systems exist based on (to varying extents) common power grid or network data of the utility. Thus, the complete model maintenance is split up in different model maintenance systems with defined data responsibilities for a specific data item. For specific parts of the data required for the Spectrum Power 7 system, one of those external systems might be the master. An example for such a system is the Geographical Information System (GIS). The data coming from the different sources needs to be consolidated by IMM into a single CIM based DOM before populating the Spectrum Power 7 runtime system._x000D_ IMM supports the consolidation of data from different sources through the ability to define the foreign identifiers to allow one-to-one identification in both, the external systems and IMM.

3.1.3 Engineering Process

The system engineering process basically consists of three phases: _x000D_ * System configuration_x000D_ * Customization_x000D_ * Data entry_x000D_ All three activities are performed during commissioning. As the requirements of the utility evolve, these activities continue to occur when the system is in operation. In particular, data entry and changes are a normal part of day-to-day system maintenance activity._x000D_ (image: 1-imgc3f3f55b6f73f9679da3523933978529_1_en_US_TIFF.jpg) _x000D_ Figure 3-2 Engineering Phases_x000D_ In general, system engineering activities follow the sequence of configuration, customization, and data entry. However, one of the key requirements of all phases of the engineering process is that the engineering tools are to accommodate an iterative philosophy of system engineering. This means that changes to previously engineered aspects of the system can be made without losing all the subsequent engineering work that has been done. _x000D_ For example, it is possible to modify system configuration information and to perform a new custom feature implementation after data entry is well underway._x000D_ System configuration and customization activities occur mainly during the initial system implementation and less frequently when the system is in operation._x000D_ How data is to be configured specifically for a certain purpose within the Spectrum Power control center system is described in detail in so called Data Modeling Guides, available for the specific application areas of the Spectrum Power control center system._x000D_ System Configuration_x000D_ System configuration comprises the following activities, where applicable:_x000D_ * Provision of hardware (servers as well as the network)_x000D_ * Installation of software (operating system, other third-party software and Spectrum Power software)_x000D_ * Basic configuration of the system_x000D_ Customization _x000D_ In this phase, project enhancements of the standard product can be introduced. The CIM based DOM can be extended to fit customer needs. Typically, data is migrated from existing systems and imported into the engineering database._x000D_ Data Entry_x000D_ Data entry describes the process of day to day changes of the data. This is done to account, for example, for changes in the physical structure of the power system network. Maintenance of the power network model is a key activity underlying the various applications in power control center systems. Keeping the static power system network model accurate and up-to-date ensures accurate and useful results within runtime operation. In particular, data entry and changes are a normal part of day-to-day system maintenance activity.

3.1.4 Domain Object Model

The Spectrum Power DOM provides a logical, object-oriented data model describing power system information, characteristics and behavior. The DOM is based on the CIM V12._x000D_ Common Information Model (CIM) _x000D_ CIM is a set of standards for representing power system components. The IEC standard 61970-301 (Energy Management System Application Programming Interface (EMS API)) that originated from the Electric Power Research Institute Control Center Application Program Interface (EPRI CCAPI) project is a semantic model that describes the components of a power system at an electrical level and the relationships between each component. The IEC 61968-11 (System Interfaces for Distribution Management) extends this model to cover other aspects of power system data exchange such as asset tracking, work scheduling and customer billing. The IEC 62325-301 (Energy Market Communication) extends both these models to cover the data exchanged between participants in electricity markets. These three key standards are collectively known as the CIM for electrical power systems._x000D_ In practical applications of the CIM it is combined with application and project specific extensions. DOM is CIM-compliant in its basic structure. It matches CIM in those parts where CIM defines information needed for the applications provided by the Spectrum Power system. _x000D_ DOM contains extensions for all parts that are not covered by CIM._x000D_ For example, communications to the physical equipment in the field, user and console function authority assignments, presentation logic assignments for field equipment in network diagrams, and so on. _x000D_ (image: 1-img63daf5336f76be009da352390a9df611_1_en_US_TIFF.jpg) _x000D_ Figure 3-3 Utilization of CIM Resources by DOM_x000D_ Extensions to the CIM V12 model are handled through the use of namespace assignments. Namespaces do not carry much information in themselves but indicate the origin of the information. The namespaces RDF and CIM are defined in CIM and delineate the CIM standards from customized extensions in DOM._x000D_ The logical structure provided by the DOM is independent of the physical database implementation and the executable software. Because the DOM hides the underlying physical implementation, it is useful as a basis for application interfaces as well as user interfaces. This de-couples a client application from the implementation details of another application that supplies data or services._x000D_ DOM Customization _x000D_ The DOM can be extended and adapted to specific customer needs. Types, attributes, and associations are assigned to namespaces. RDF is the base namespace, it is extended by CIM. Both namespaces are defined in the CIM standards. DOM extensions for the Spectrum Power domain have an own namespace defined by the product. Customer-specific DOM extensions finally get a customer-specific namespace._x000D_ (image: 2-imgd73956fb6f799efe9da352391cdf2b4e_1_en_US_TIFF.jpg) _x000D_ Figure 3-4 Spectrum Power Domain Object Model – DOM_x000D_ Using namespaces allows additional, extended data to be integrated with the standard CIM without affecting the general information model definitions. Namespace assignments make it easy to maintain compatibility with CIM standard-based interfaces. _x000D_ For example, interfaces can correctly identify standard CIM data from the extended, non-standard data and can either choose to ignore or consider this data.

3.1.5 Functions

Spectrum Power IMM controls the data to be defined and transferred between the engineering database and the Spectrum Power runtime databases._x000D_ IMM provides functions that act like a set of tools to maintain power system information. The sub-functions of IMM are as follows: _x000D_ * Job management_x000D_ * Domain and graphics model data maintenance_x000D_ * IMM trigger framework_x000D_ * Data import and data export_x000D_ * Validation of data changes_x000D_ * Activation of data changes_x000D_ * Data version management_x000D_ * Reporting_x000D_ * Logs_x000D_ * Multisite environment support_x000D_ * Quality Assurance System (QAS) support_x000D_ * Operator Training Simulator (OTS) support_x000D_ * Access rights_x000D_ * IMM administration

3.1.6 IMM Data Definition

The Spectrum Power IMM functions are a set of tools that allow power system information data to be defined, accessed, and exchanged. These tools also control the transfer of data between the engineering database and the Spectrum Power runtime databases._x000D_ The propagation of data changes from the IMM to the Spectrum Power applications online runtime databases is designed so that only incremental changes are applied to these databases. The application of the changes to the runtime databases is coordinated so that there is no application downtime – this includes the process interfaces and real-time applications. All changes are immediately available for processing and viewing by operators, and there is no interruption in the real-time operation of the network control center. This process is known as Online Activation._x000D_ (image: 1-imgc85e1a566f7c44469da352390cb056a1_1_en_US_TIFF.jpg) _x000D_ Figure 3-# Spectrum Power IMM Functional Overview_x000D_ * Engineering data import and export can be done in XDF and RDF formats based on the W3C standard XML._x000D_ * Partial power grid/network data export can be done additionally in Comma-separated Values (CSV) format._x000D_ * Editing engineering and graphics data._x000D_ * IMM Editors used for editing engineering data and Graphics Editor used for editing graphic network diagram data._x000D_ * Validation features provided by IMM ensures that the data model remains consistent._x000D_ * Reporting features provided by IMM allow the user to create and view summary or detail reports of type and instance data._x000D_ * Job management is the method by which changes of the Spectrum Power engineering database are grouped and managed in jobs._x000D_ * The engineering data is stored in Oracle source data base and go through data preparation and data activation phases to get the changes applied._x000D_ * Data version management and automatic data model archiving facilities provides a history of model changes including auditing capabilities.

3.1.7 IMM Engineering Applications

IMM has a number of engineering applications suitable for the different engineering tasks._x000D_ (image: 1-img65d5925470b687369da352393f7fa3ae_1_en_US_TIFF.jpg) _x000D_ Figure 3-# General Application Structure of the IMM User Interface_x000D_ The amount of provided applications is dependent of the concrete customer project and configuration._x000D_ IMM Application provide the following: _x000D_ *

Type Editor_x000D_ The Type Editor is used to view and edit properties of a type as well as to create new types. It provides a set of tabbed pages each of which is used to configure a distinct kind of type properties._x000D_ * Type Inspector_x000D_ The Type Inspector is used to view type properties. Basically it is the read-only version of the Type Editor._x000D_ * Model and Graphics Editor_x000D_ The Model and Graphics Editor allows viewing and editing of network model and diagram data. It provides structured panels dedicated to different views to the data and rich navigation means through the different views._x000D_ * Multi-Instance Editor_x000D_ The Multi-Instance Editor is used to view and modify a set of instances including links as well as to search and filter for instances. Queries can be defined, saved and loaded for reuse. The result table can be exported into a CSV file._x000D_ * Symbol Editor_x000D_ The Symbol Editor allows viewing and editing symbols used on the diagrams._x000D_ * Color Editor_x000D_ The Color Editor allows viewing and editing color values of color instances used on the diagrams._x000D_ * Decision Table Editor_x000D_ The Decision Table Editor allows maintaining diagram decision tables for evaluating the presentation of dynamic display objects in the runtime environment._x000D_ * Shape Style Editor_x000D_ The Shape Style Editor allows creating reusable shape styles which represent a particular graphic property combination that can be assigned to graphic objects._x000D_ * Style Group Editor_x000D_ The Style Group Editor allows maintaining unique styles matching the values configured in the diagram decision tables by the Decision table editor in a certain style group._x000D_ * Style Logic Editor_x000D_ The Style Logic Editor allows maintaining style logics for evaluating the style of the presentation of dynamic display objects in the runtime environment based on the combination of a decision table and a style group._x000D_ * Symbol Group Editor_x000D_ The Symbol Group Editor allows maintaining unique symbols matching the values configured in the diagram decision tables by the Decision table editor in a certain symbol group._x000D_ * Symbol Logic Editor_x000D_ The Symbol Logic Editor allows maintaining symbol logics for evaluating the symbol of the presentation of dynamic display objects in the runtime environment based on the combination of a decision table and a symbol group._x000D_ * Text Style Editor_x000D_ The Text Style Editor allows creating reusable text styles which represent a particular graphic property combination that is assigned to graphic text objects._x000D_ * Analog Representation Editor_x000D_ The Analog Representation Editor allows viewing reusable analog representation styles._x000D_ * Job Management_x000D_ The Job Management allows the user to control all domain and graphical data entry and engineering activities of the power system._x000D_ * Import dialog_x000D_ The Import dialog allows the user to import type and instance data from an XML file. The supported file formats are shown in the dialog._x000D_ * Export dialog_x000D_ The Export dialog allows the user to export type and instance data into a file. The supported file formats are shown in the dialog. Additional export options can be selected._x000D_ * Reports_x000D_ Report features allow the user to create and view summary or detail reports of instance data._x000D_ * Instance Search dialog_x000D_ The Instance Search dialog allows the user to search for instances based on search criteria._x000D_ * Logs_x000D_ Logs provide important information during the engineering process. Selected logs can be exported.

3.1.8 IMM UI Technology

A typical data engineering console consists of multiple monitors. During an IMM engineering session, the console is connected to the IMM server running on

Administrator Server (ADM). Multiple engineering consoles can be connected to the IMM server._x000D_ The IMM UI client program can be installed on any console in the control center. It runs on Windows or Linux. The client is updated as needed automatically upon login. The initial window loaded is the main screen with the job management user interface and serves as the general access point to the data engineering jobs and IMM engineering applications._x000D_ The IMM UI client communicates with IMM Server over a secure Message Interchange Communication (MIC) connection. The ADM can run in a spare redundancy configuration. In case of a failure and run up of the spare server, the IMM UI client connection has to be re-established manually._x000D_ (image: 1-imgc9559f22b3235aaf9da35239753dbfa5_1_en_US_TIFF.jpg) _x000D_ Figure 3-# IMM UI Basic Architecture_x000D_ (1) IPC : Inter Process Communication_x000D_ (2) IMMUIDP: Information Model Management User Interface Data Provider_x000D_ (3) MIC: Message Interchange Communication

3.2 Job Management

3.2.1 Generals

Domain data, graphical network diagram data entry, and engineering activities are under the control of the IMM job management. Job management is the method by which changes of the Spectrum Power engineering database are grouped and managed. A job allows multiple and concurrent users to modify data simultaneously in a secure environment, without impacting the operational databases of the Spectrum Power runtime system._x000D_ Case Study: Introduce a new substation and propagate this substation to the runtime system._x000D_ * Create a job in IMM_x000D_ * Edit the new substation domain data, topology and graphical data in the job_x000D_ * Validate the job_x000D_ * Activate the job_x000D_ * Verify the job history_x000D_ Job management controls the definition, usage and deletion of jobs. _x000D_ Job management provides the following functions:_x000D_ * Defining a new job or continuing to work within an existing job. Multiple jobs can exist at one time._x000D_ * Associating data changes with a specific job._x000D_ * Viewing data changes associated with a job._x000D_ * Viewing the job list together with the job status._x000D_ For example, job reservation details, whether a job has been validated or activated, and so on._x000D_ * Validation of complete data or incremental data changes in a job._x000D_ * Activation of data changes to the Spectrum Power runtime system._x000D_ * Undo of activated data changes to the Spectrum Power runtime system._x000D_ * Archiving of job changes after a successful activation of the runtime system – if enabled._x000D_ * Erasing a job discarding all data changes associated with the job._x000D_ * Finalizing the job to make changes permanent._x000D_ Data entry and engineering activities include the following:_x000D_ * Manual data entry_x000D_ * Import of bulk or incremental changes from external data model or instance data_x000D_ * Generating change log reports_x000D_ Job_x000D_ A job groups data changes that belong together. The job is the unit of data that is finally changed in the engineering database and in the runtime databases of the Spectrum Power system. Each job is identified by a name. Although the job management function provides the capability to group data changes into a job, it is the data engineer who is in control of grouping the changes together. This ability adds great flexibility to the data engineer as to what data changes define an increment of change against the database

currently in operation._x000D_ (image: 1-img084eb9fe70c4938c9da3523935bd81d0_1_en_US_TIFF.jpg) _x000D_ Figure 3-# Job Concept_x000D_ The figure uses a collection of cubes to represent data instances of the production model (production model is the model used in the Spectrum Power runtime system) and in a job. _x000D_ The arrows indicate your view of the data. Arrow A depicts what is viewed when an instance was modified in a job. Arrow B indicates what is viewed when an instance is unchanged. _x000D_ A job follows a defined lifecycle and can adopt different states according to the current workflow. _x000D_ (image: 2-img2a41753970b133fd9da3523966801ed6_1_en_US_TIFF.jpg) _x000D_ Figure 3-# Life-cycle Management for Data Modifications within an IMM Job_x000D_ After a job is created, data can be edited. After the data changes are done within the job, data/job should be validated for correctness and completeness. Preparation prepares data for the runtime system. Activation makes prepared data available to the Spectrum Power runtime system and its applications._x000D_ Job Reservation_x000D_ Multiple jobs can exist simultaneously. Several users can work on different jobs at the same time. A single job is reserved for a particular user. When a user reserves a job, only that user can work with it, other users will only have access to view the data in this job. The current job owner and an authorized user can reassign a job to a different user._x000D_ Change Detection_x000D_ As the user changes data in a job, the incremental changes are detected and recorded in the change log. The change log contains modifications of instances and links. Multiple changes to the same data item in the job due to multiple updates are merged. The change log provides a means to view what was changed in the past and can be used for auditing purposes. It is used also by the activation process to determine the changes to be processed into the operational database of the runtime system. _x000D_ Job History Log_x000D_ The job history log contains all status changes of a job, the date and time of the change and the user who made the status change. This includes job owner reassignments.

3.2.2 Independent Job Mode

In independent job mode, you can view or edit the model as it is at the current time when you are in a job. You see the model as it currently is (the production model used in the Spectrum Power runtime system) plus your job changes. You do not see changes from other jobs unless the jobs are activated. The changes in the job can be considered as overlays on top of the data currently in use online. Each job is a separate overlay so switching between jobs shows the changes in the job overlaying the current data in use. _x000D_ (image: 1-imgdaab9ce7710fefab9da35239293d523b_1_en_US_TIFF.jpg) _x000D_ Figure 3-# Independent Job Mode View (1)_x000D_ Job interlocking prevents different jobs from modifying the same instance and link data to preserve data integrity. This means, that you cannot change the instance data that has been locked by another job and other jobs cannot change your data._x000D_ (image: 2-imgb42e945470b6898c9da352391a5e23e7_1_en_US_TIFF.jpg) _x000D_ Figure 3-# Independent Job Mode View (2)_x000D_ For example, two different jobs (B and C) cannot edit the same part of equipment within a substation. If you try to change data locked by another job, an error message is issued indicating that the data is locked and your change is aborted. _x000D_ This keeps jobs in the engineering environment independent from each other. They can be processed (for example, activated, undone, or finalized) in any order. _x000D_ Job Interlocking_x000D_ Job interlocking applies to individual instances or links in the data model. Adding, modifying or deleting an instance

automatically locks the instance itself only. When the user changes an attribute value of an existing instance, this object is locked by the job. If the name of the object is changed, both the old and new names are locked. Both, parent and child objects are not affected. When a new object is added, the new instance is locked. Only this job can create descendants to the newly created instance._x000D_ When an object is deleted, all child instances are also deleted, and all the deleted instances are locked. Another job cannot insert a new instance with the same name as the deleted instance._x000D_ The deletion of an instance is prevented if one of the descendants of their links is locked by another job to ensure data integrity._x000D_ Adding or deleting a link locks this link only. This means that the exact same link, that is, a link that references the same two instances using the same association, cannot be added respectively deleted again by another job._x000D_ After a job has been activated and finalized or deleted, all interlocks created by this job are released. _x000D_ Job Interdependency Checks_x000D_ Job interlocking is performed at engineering design time and supports the user’s data maintenance with fine granular individual instance locking, thus allowing utmost parallel editing work without blocking each other. IMM also makes sure that the jobs to be activated are distinct from each other regarding the Spectrum Power runtime databases. Interdependency checks are performed during data preparation. Preparing a particular job is possible if the job changes are independent of any of the previous still prepared, transferred or activated jobs. This check is based on the runtime models to ensure data integrity in terms of the Spectrum Power runtime system.

3.3 Domain Data Maintenance

Overview_x000D_ Engineering activities to change data require working with large amounts of information with multiple attributes and properties. IMM is the user interface for domain data maintenance within a job. _x000D_ Domain data editors provide means for the following:_x000D_ * Instance data modifications_x000D_ * Data type changes_x000D_ These are provided in different but consistent views of the data according to your intention and workflows._x000D_ Creation and Management of Instances_x000D_ The following instance data modifications can be done from IMM UI within a job: _x000D_ * Create instances_x000D_ * Rename instances_x000D_ * Modify attribute values for instances_x000D_ * Delete instances_x000D_ All descendent instances and their links are also deleted._x000D_ * Create links_x000D_ * Delete links_x000D_ Type Changes_x000D_ Data model maintenance and customizing features allow for the extension of the standard DOM by creating and maintaining custom types. IMM provides means to extend and manipulate the DOM. _x000D_ The following actions are possible:_x000D_ * Creating, modifying, and deleting types._x000D_ Deletion of types deletes all derived types. A type can be deleted only if all instances of this type have been deleted beforehand._x000D_ * Creating, modifying, and deleting attributes of a type._x000D_ Derived types inherit all attributes of its base types._x000D_ * Creating and deleting associations._x000D_ DOM modifications can be either performed with the Type Editor or by import of XDF files._x000D_ When a type change is made, all existing jobs and all subsequent jobs inherit the type change. Since type changes affect the current data model and future jobs, only a user with administrative rights is allowed to make type changes.

3.4 Graphic Data Maintenance

Overview_x000D_ Display construction of network diagrams is completely integrated in the IMM. The Graphics Editor provides means to view, create and modify graphic diagrams and also symbology. The graphical editing creates the link between the instances of the graphic data to instances of the domain data._x000D_ Technological Editing_x000D_ Technological editing allows creation, modification, and deletion of electrical power system domain data and topology in a graphical way. _x000D_ The following workflows are supported: _x000D_ * Create diagrams to represent power network/system graphically._x000D_ * Create new graphical objects and new domain data._x000D_ * Link graphical objects to existing domain data._x000D_ * Create and manipulate the topology of the electrical power system network._x000D_ * Delete domain data using deletion of graphic objects linking the domain data._x000D_ * Support creation and usage of user-defined templates that contain domain data and graphic data._x000D_ Technological editing maintains consistency between graphic data and electrical domain data._x000D_ Raster Set Assignment_x000D_ Generally, rasters are digital aerial photographs, imagery from satellites, digital pictures, or even scanned maps. Picture raster data is often used in geographic displays to convey additional information. Raster data might be organized in raster datasets and raster catalogs. The raster set assignment allows assigning installed raster sets to layers of a diagram and supports PNG and JPG formats.

3.5 IMM Trigger Framework

IMM Triggers execute a set of business logic required by downstream applications as it applies to the data. The IMM Trigger functionality provides a user-friendly data entry support. Trigger functions are able to perform actions not only based on an insert, update, and or delete of instances, but also based on current values and situations existing in the database. _x000D_ For example, having triggers that auto-create required instances either as child objects or objects elsewhere in the domain data hierarchy based on certain data criteria, means that there is less data that the user must enter by hand and less chance for human error._x000D_ Trigger Events_x000D_ Triggers are executed based on a trigger event. Trigger events are part of the data model type definitions in DOM. _x000D_ There are five types of events possible. They are as follows:_x000D_ * OnInsert_x000D_ * OnUpdate_x000D_ * OnDelete_x000D_ * OnCutPaste_x000D_ * OnRename_x000D_ The trigger functions are executed after the end of an engineering task. _x000D_ For example, if one task consists of adding three instances of type Breaker and deleting one instance of type Measurement, the trigger function assigned to OnInsert Breaker event will be called three times, once for each Breaker, and the function assigned to OnDelete Measurement will be called once for the Measurement. All necessary trigger functions are called after changes are made by the task and are completely committed to the database._x000D_ Trigger Functions _x000D_ Triggers are data type dependent. They reference types defined by the data model DOM. One generic master trigger package, PkgTrgMaster, resides in the schema directory and handles the calling of all necessary trigger functions. _x000D_ The following trigger functions are supported:_x000D_ * Population of internal (read-only) attributes for instances_x000D_ For example, auto-numbers are needed to support the Spectrum Power runtime system like B1, B2, B3, Diagram, and Nimset numbers. The auto-number defined for an instance A is unique when associated with another instance B. The instances A and B may be a parent child or a reference association. A defined set of associations may drive a

certain auto-number. The numbers can be configured to be Reused or Non-reused._x000D_ * Creation of children or grandchildren instances_x000D_ Auto-create new child objects when a new instance is added._x000D_ For example, creation of Info’s under elements based on the normelement type, creation of data acquisition equipment, creation of RTDS pseudo station, and so on._x000D_ * Population of default values of certain attributes for instances_x000D_ Triggers are capable of populating an attribute with a dynamic default. A dynamic default is a default value that is generated based on the current values in the database and the data being entered at the time it is assigned._x000D_ For example, the LogbookText attribute. By default, this is the same as the name of the object, but it can be override by specifying an actual value._x000D_ Extension of Trigger Functions _x000D_ Trigger functions can be run as part of the trigger framework. They can be implemented in Procedural Language or Structured Query Language (PL/SQL). The trigger framework component can be accessed using an external customer application using the XDF.

3.6 Data Import and Data Export

3.6.1 General

Import and Export of Engineering Data in XDF or CIM-RDF _x000D_ IMM provides an interface to export and import engineering data in XDF and CIM-RDF. Both are XML formats based on W3C standard. The XML is a versatile language for the definition of tags to identify document contents. XML allows third-party products to be used for data editing and pre-validation._x000D_ CIM-RDF is based on the IEC 61970-452 and IEC 61968-13 standards for the description of electrical power systems. XDF is a proprietary format. IMM supports complete, partial, and incremental import and export in XDF and CIM-RDF. All engineering data can be exported and imported in these formats. _x000D_ Profiles allow filtering of exported engineering data. They are based on types, attributes, and associations. You can define types, attributes, and associations that define a profile. An arbitrary number of profiles are supported._x000D_ Instance filtering of exported engineering data is based on logical expression containing one or multiple attributes of one type. Instance filtering is provided to allow limiting the exported instances to a set matching a certain use case._x000D_ The data import and export features of Spectrum Power IMM are used for:_x000D_ * Data migration_x000D_ Transfer of instance data and graphics data from legacy systems._x000D_ * System upgrade_x000D_ The import or export feature allows you to store the database content in files that are re-imported after system upgrade._x000D_ * External engineering data modification_x000D_ The import/export feature allows modifying data in external files which, after external entry modification, are re-imported into the IMM data model._x000D_ * Data backup feature_x000D_ The export function can be used to save the data (the overall data model) to an external file._x000D_ * Profile-based export_x000D_ Only instances assigned to a certain profile are exported._x000D_ Following the data import, validation ensures that the new engineering data is consistent and does not conflict with any other engineering data in the system. During export, the specified data information is transferred from the database to external files._x000D_ Further Instance Data Export Formats_x000D_ Partial power grid/network data export in CSV format is supported._x000D_ In power companies, several systems exist based on (to varying extents) common power grid/network data of the utility.

Generally, each system has a database and its own data model maintenance tools optimized to the specific scope of the data. Thus, the complete model maintenance is split up in different model maintenance systems with defined data responsibilities for a specific data item._x000D_ For specific parts of the data required for the Spectrum Power system, one of those external systems might be the master. The data coming from the different sources needs to be consolidated by IMM into a single source before populating the Spectrum Power runtime system. _x000D_ An example for such a system is the GIS. Data provided by the GIS gets imported into IMM through the GIS Data Import Management (GDIM) function of Spectrum Power._x000D_ The following feature of IMM support the consolidation of data from different sources in the CIM based DOM:_x000D_ * Support of foreign identifiers_x000D_ The CIM based DOM in IMM uses, among others, universally unique RDF identifiers. External systems might use foreign unique identifiers (for example, GIS ID) for similar purposes. Foreign identifiers can be defined as such in DOM and allow for one-to-one identification in both systems. Foreign identifiers are treated similarly to RDF identifiers in IMM regarding enforcement of the uniqueness during modification attempts.

3.6.2 Model Merge Framework

In power companies, several systems exist based on (to varying extents) common power grid/network data of the utility. Generally, each system has a database and its own data model maintenance tools optimized to the specific scope of the data. Thus, the complete model maintenance is split up in different model maintenance systems with defined data responsibilities for a specific data item._x000D_ For specific parts of the data required for the Spectrum Power system, one of those external systems might be the master. The data coming from the different sources needs to be consolidated by IMM into a single source before populating the Spectrum Power runtime system. _x000D_ An example for such a system is the GIS. Data provided by the GIS gets imported into IMM through the GIS Data Import Management (GDIM) function of Spectrum Power._x000D_ The following feature of IMM support the consolidation of data from different sources in the CIM based DOM:_x000D_ * Support of foreign identifiers_x000D_ The CIM based DOM in IMM uses, among others, universally unique RDF identifiers. External systems might use foreign unique identifiers (for example, GIS ID) for similar purposes. Foreign identifiers can be defined as such in DOM and allow for one-to-one identification in both systems. Foreign identifiers are treated similarly to RDF identifiers in IMM regarding enforcement of the uniqueness during modification attempts.

3.7 Validation of Data Changes

Validation ensures that the entire data model remains consistent. In addition, it ensures that all necessary data is entered (completeness check). Validation takes place in a maintenance environment, for example, a job, before the changes are activated into the Spectrum Power runtime system. Validation is started either user triggered using the IMM user interface or automatically when saving data that has been entered using an IMM editor._x000D_ _x000D_ (image: 1-Tip.png) | NOTEIt is recommended to run validation and fix constraint violations before activation._x000D_ Validation Areas_x000D_ Validation can be performed for the following areas:_x000D_ * The data changed in a job_x000D_

* Entire data in the root combined with the data changes in a job (global validation)_x000D_ * Data modified by means of the IMM import_x000D_ Validation Types_x000D_ The following validation types can be distinguished:_x000D_ * Inter-object validation_x000D_ Inter-object validation refers to those checks that require multiple instances to ensure data consistency. These checks can also include instances of different types._x000D_ * Intra-object validation_x000D_ Intra-object validation verifies the correctness of data in a single instance. This is done automatically when the data is saved. If an incorrect data value is entered, an error message is issued. Intra-object validation includes:_x000D_ * Unique value checks_x000D_ These checks verify that the value in an attribute or combination of attributes is not duplicated across multiple instances of an object._x000D_ * Cardinality checks_x000D_ These checks verify that relationships between types are satisfied._x000D_ For example, a parent or child relationship._x000D_ * Required attribute checks_x000D_ These checks verify that a value has been supplied for a required attribute._x000D_ * Range checks_x000D_ These checks verify that the value is greater than or equal to a minimum value and less than or equal to a maximum value._x000D_ For example, 1-10._x000D_ * Enumeration checks_x000D_ These checks verify that a value is member of a specified list of values (enumeration)._x000D_ For example, Yes/No, Open/Close._x000D_ * Special validation checks_x000D_ These checks verify that a special condition is satisfied. Special validation checks are dependent on the data being validated._x000D_ For example, if attribute A is Yes, then attribute B is required._x000D_ * User-defined validation_x000D_ User-defined validation provides the ability to create specific validation checks, if required, and the ability to integrate these additional checks into the validation process. A validation rule calls a specific validation script and specifies appropriate parameters. The validation script is the code behind the rule. Each script can be called by many rules depending on the nature of its definition. Simple rules can be defined in XML, complex rules are implemented in PL/SQL._x000D_ The data validation component can be accessed through an external customer application using the XDF._x000D_ Validation Report_x000D_ The validation report provides text messages from the server during a save operation to the database. It is also provided for some other operations such as creating diagrams using a profile or using templates._x000D_ Validation Log_x000D_ The validation log provides a summary of all validation messages generated during the validation of a job.

3.8 Activation of Data Changes

All power grid domain data and diagram data changes are done in a job. Activation propagates data changes into the Spectrum Power runtime system. The activation ensures that the incremental changes are applied to all applications of the Spectrum Power runtime system, including IMM. In a single failure during activation, the changes are rolled back system-wide. _x000D_ Activation_x000D_ Activation is performed in three phases:_x000D_ * Data Preparation_x000D_ Data Preparation generates the change log files for the different application suites, populates Spectrum Power object tables and creates and populates FxBin files for network diagram data for the incremental changes associated with a job._x000D_ Multiple jobs can be in the prepared state if they are distinct from each other in terms of the operational database of Spectrum Power. Interdependency checks during data preparation are performed to check whether the changes of the job are distinct to all jobs being in prepared, transferred, or activated state. This check is based on the runtime

models to ensure data integrity._x000D_ After a successful data preparation, the incremental changes are ready to be transferred into the offline copy of the operational database._x000D_ * Data Transfer_x000D_ Data Transfer populates the incremental changes associated with a job to the offline copy (copy 1) of the operational database of Spectrum Power. With this step the Spectrum Power ADM gets involved for the first time._x000D_ After a successful data transfer, the incremental changes are ready to be activated into the online copy of the operational database._x000D_ * Data Activation_x000D_ In this phase the incremental job changes are merged into the root of the dataset in the IMM database. Data Activation activates the changes into the online copy (copy 0) of the operational database of Spectrum Power and into the ASRs of Shared Components._x000D_ Undo Activation_x000D_ An Undo Activation is considered as rare and exceptional use-case. Nevertheless, Undo Activation might be the fastest way to return back to normal runtime operation when logically correct, thus without validation errors, but faulty data not describing the real world got activated. Instead of correcting the faulty data under time pressure in a new job, undo activation provides convenient means to undo the changes without losing the former data changes that causes the runtime failure. The faulty data can be corrected in the original job and activated again. _x000D_ Undo Activation has the following two phases:_x000D_ * Undo Prepare_x000D_ In this phase, an undo change log is generated for the Shared Components (like HIS) and deployed. For Spectrum Power applications, this phase is a no-op._x000D_ * Undo Activation_x000D_ The changes are undone from all application suites, including IMM. For Spectrum Power, the changes are undone from copy 0, copy 1 and the job changes are removed. The job status is set back to Edit._x000D_ Finalize Activation_x000D_ After finalization, changes are made permanent and the job gets deleted within the Job Management UI and can no longer be undone. Associated job locks are released.

3.9 Data Version Management

Data version management and automatic static data model archiving facilities provide a history of model changes and allows the user to track data changes over time. Jobs in the IMM model archive provide a past view of the static data model based on the activation time. If archiving is enabled, data is stored in the IMM model archive after a successful job activation or undo activation of the Spectrum Power runtime system. This allows the user to view all jobs that have been activated in the past. The user can run change log reports on archived jobs, access logs, export data, and export data changes in the same way as it is done in the current Real-Time Dataset (RT). The active IMM model archive is a separate read-only dataset; data editing or import is not allowed in order to preserve the history._x000D_ Retrieving a past model for any point in time enables accurate conclusions to be derived from post-mortem analysis._x000D_ DOM Type Changes_x000D_ Special consideration is made when changing DOM types. Simply applying type changes to the active archive would distort the way instance data looked in the past. However, not applying type changes would mean that future jobs could not be archived. Therefore, type changes that do not affect the instance data are applied directly to the active archive. Type changes that affect the instance data are logically applied. The changed attribute determines which changes are made to the archived instances._x000D_ Long Term Archive_x000D_ Creating a new baseline in the active archive allows the user to transfer a specific historical time frame into a long-term archive. A long-term archive is an oracle dump file that can be stored on external media and can be loaded again into a dataset for any

purpose._x000D_ (image: 1-imgde5c8cd170e558639da35239674b056f_1_en_US_TIFF.jpg) _x000D_ Figure 3-# Spectrum Power IMM – Model Archive_x000D_ Auditing of IMM Jobs_x000D_ IMM job auditing features provide means to keep track of who made when what changes to the domain and graphical data. _x000D_ For example, when a problem is identified for a given data instance, end users get the ability to identify what jobs made changes to that instance. Auditing features require the jobs being in the active IMM archive. The specific provisions are as follows:_x000D_ * For any job that is finalized, you can export the incremental XDF or RDF change log and view the report of the changes in that job._x000D_ * The user is able to view the previous value and the new value changed to by the job._x000D_ * The user is able to view the order in which jobs were activated or undone and when they were finalized._x000D_ * The user is able to identify the jobs that modified a given instance within a given time frame.

3.10 Reporting

Data Reporting _x000D_ Reporting features provided by IMM allow the user to create/view summary or detail reports of type and instance data._x000D_ Instance Change Report_x000D_ The instance change report displays changes within a selected network equipment hierarchy and within any hierarchy below substation like voltage levels or bays. The report contains all changes made in the existing jobs of a dataset to the instance hierarchy. In the independent job mode, the currently opened job in the real-time dataset and the root job are evaluated._x000D_ Object Usage Report_x000D_ The object usage report displays for a selected network equipment hierarchy the linked telemetry data points as well as linked graphical objects in the different network diagrams. The report can be created for substations or any hierarchy below like voltage levels or bays. The telemetry information covers the linked Independent Front-End System (IFS) that is, RTU-Front-End data as well as linked Inter-control Center Communications Protocol (ICCP) data.

3.11 IMM Logs

IMM provides logs within the log section of the IMM user interface. The log section can be opened in a separate window. Selected logs can be exported to a Comma-separated Values (*.csv) file. _x000D_ The following are the various log types available, depending on the selected dataset, job or data engineering workflow:_x000D_ * IMM Status Log_x000D_ Messages about the state (activation, preparation, and so on) of the job or IMM itself. The messages include severity, job name, time stamp and source._x000D_ * Task Log_x000D_ Messages about user actions such as copy/paste, cut/paste, delete subtree, modify instance data, modify type data, and so on. The messages include severity, details, time stamp, and source. An object path is a hyperlink that can be used to navigate to the instance in question._x000D_ * Validation Log_x000D_ Messages about job validation or global validation. The messages include severity, hierarchical path of the instance and the object type. The object path is a hyperlink that can be used to navigate to the instance in question._x000D_ * Activation Log_x000D_ Contains all the messages that were created during the preparation or activation of a dataset or job. The messages include severity and time stamp._x000D_ * Import Log_x000D_ Contains all the

messages that were created when a user initiated import was done within a dataset or job. The messages include severity, details and time stamp._x000D_ * Job History Log_x000D_ Shows the status change of the selected job. The messages include last status change, job name, job status, previous job status and user._x000D_ * Finalized Jobs Log_x000D_ Lists the jobs that were activated and finalized. The messages include last status change, job status, previous job status and user._x000D_ The individual log can be modified. Predefined columns with additional information can be included; columns can be used for sorting or can be hid._x000D_ In the active IMM model archive, the Activation Log and Job History Log for the jobs are retained for auditing purposes.

3.12 Spectrum Power Operating System

3.12.1 Multisite Environment Support

The collection of control centers cooperatively managing a power system are known as a multisite system. Multisite systems are usually organized in a main/backup or main/regional configuration. All Spectrum Power systems in a multisite network have the complete data model with formal identical content of information maintained by IMM. There is only one data model master at a time. The Main Control Center takes the role of the data model master. The other control centers of the network are source data-dependent._x000D_ Main/Backup Control Center_x000D_ Systems having Backup Control Centers require synchronized engineering databases between the Main and Backup Control Centers. Data consistency (editing and activating) between the two sites is guaranteed using Oracle data guard and the IMM activation framework._x000D_ (image: 1-img9f35d3c670c495229da35239137c7662_1_en_US_TIFF.jpg) _x000D_ Figure 3-# Main/Backup Configuration – Basic Overview_x000D_ When there is a switchover or fail-over, authorized data engineers can set the current Main Control Center as master for IMM at any time._x000D_ Main/Regional Control Center_x000D_ Regional Control Centers get their source data changes applied through job activation from the Main Control Center. There is no IMM database at a Regional Control Center, thus data engineering is always performed on its Main Control Center._x000D_ (image: 2-img545512f770c496ac9da352390056d58b_1_en_US_TIFF.jpg) _x000D_ Figure 3-# Main/Regional Configuration – Basic Overview

3.12.2 Quality Assurance System (QAS)

A QAS allows testing data changes without any implication to the production system. The production system and QAS are independent from each other. The QAS takes the role of the Data Model Master. Modified and successfully tested IMM data is transferred from QAS to the production system only. Activations of the production system are always triggered on the IMM UI at the QAS._x000D_ (image: 1-img4dc87ebc70e553ed9da352390f74a9bc_1_en_US_TIFF.jpg) _x000D_ Figure 3-# QAS System Configuration – Basic Overview_x000D_ As long as the QAS is running and is reachable by the production system, domain data modification is not allowed in IMM on the production system. If the QAS is

down, authorized data engineers are enabled to set the production system as master for IMM at any time._x000D_ The mandatory IMM job business model in a QAS system configuration is the independent job mode._x000D_ If the QAS runs up the first time a full IMM database synchronization is done. If jobs already exist at the production system these are synchronized as well.

3.12.3 Operator Training Simulator

An Operator Training Simulator (OTS) enables operators to practice runtime system operations under simulated conditions. The main system and the offline OTS are independent from each other._x000D_ (image: 1-img570ef35971345bbb9da3523925336202_1_en_US_TIFF.jpg) _x000D_ Figure 3-# OTS System Configuration – Basic Overview_x000D_ If the OTS runs up the first time a full database synchronization is done with its main system.

3.13 IMM Access Rights

The Data Engineering (DE) comprises all tools related to the provision of engineering and parameter data to the Spectrum Power system (during commissioning and subsequent modifications/extensions). The main modules of Data Engineering are the Spectrum Power GIS Data Import Management (GDIM) and the Information Model Management (IMM) component.

User authorization is performed during log on to Spectrum Power IMM. IMM access rights and instance level access rights are configured within the user administration dataset. By default, a user who is authorized to use Spectrum Power IMM is permitted to view the information available within IMM.

3.13.1 IMM Access Rights

Data entry and activation in IMM is controlled by access rights. IMM provides granular access rights dependent on the dataset and the requested action._x000D_ The following individual access rights are supported:_x000D_ * Instance data engineering_x000D_ * Type data engineering_x000D_ * Data activation_x000D_ * Engineering dataset administration_x000D_ * System management information instance data engineering_x000D_ * User Administration (UA) instance data engineering

3.13.2 Instance Level Access Rights

Access rights can be assigned for each instance individually. They describe what a user is allowed to do with the respective instance in IMM (view, modify, modify and assign new access rights for this instance). Instance level access rights, define on what parts of the power network data model in RT, the user can do modifications. Thus, they limit the user's given IMM access rights.

3.13.3 IMM Console Access Rights

Console access rights allow for location-based access control based on the IMM UI server (console) where the user currently is working. The authorities are always calculated as intersection (common subset) of access rights for console and user. Thus, granted IMM user access rights can be restricted by IMM console access rights.

3.13.4 IMM Job Reservation

A single job is reserved for a particular user during its creation. The current job owner and an authorized user can reassign a job to a different user.

3.14 IMM Administration

IMM Admin Command Line Tool_x000D_ The IMM admin tool is used to manage the Spectrum Power IMM datasets._x000D_ Managing Datasets_x000D_ The IMM admin tool provides authorized users the following functionality:_x000D_ * Creating a new dataset (planning dataset)._x000D_ * Clearing all instance data of a dataset._x000D_ * Clearing all type and instance data of a dataset._x000D_ * Managing the IMM model archive._x000D_ * Managing the QAS connection._x000D_ * Synchronizing ODB-Relations._x000D_ * Changing dataset properties._x000D_ For example, definition of the maximum import errors after import gets aborted._x000D_ Archive Management _x000D_ The IMM admin tool allows managing the IMM model archive. It allows the user to enable or disable the archive, to create a long-term archive and to create a new baseline in the active archive. The archive log shows details, such as the timeline where a certain data model version was used at the production system (start date and time, end date and time).

3.15 IMM User Interface

3.15.1 Generals

The IMM UI can be installed on any UI console. IMM UI runs on Windows or Linux. The client is updated as needed automatically upon login. Within the user interface of IMM, multiple editors that are optimized for the various data engineering workflows allow data access and definition covering all aspects of data engineering._x000D_ The following functions are available in IMM:_x000D_ * Creating types_x000D_ * Creating instances_x000D_ * Creating instances by copying existing instances_x000D_ * Deleting types and instances_x000D_ * Viewing, creating, and modifying attributes and properties of types and instances_x000D_ * Viewing, creating, and deleting associations between types_x000D_ * Viewing, creating, and deleting links between instances_x000D_ * Validating data changes_x000D_ * Activating data changes_x000D_ * Importing and exporting type and instance definitions_x000D_ * Creating reports of types and instance definitions_x000D_

* Creating and modifying graphic diagrams

3.15.2 Search Function

The search function allows looking up instances by the instance name or parts of the instance name. Entering the name of a parent instance narrows the search range down to the descendants of the selected instance. Placeholder characters can be used to extend the search range. Searching instances based on provided RDF ID, object identifier (GUID) or Spectrum Power Internal Identifier (SIID) is also provided._x000D_ The result of a search is a list of instances identified by their path. The user can navigate to an instance in the list by double-clicking the path (hyperlink)._x000D_ (image: 1-img6bcf7675a63478ad9da35239461ae501_1_en_US_PNG.png) _x000D_ Figure 3-# Search Function - Instances_x000D_ The search function allows searching for instances by B1/B2/B3/Elem/Info names and numbers as well._x000D_ (image: 2-img7dca5ea1a6346f749da35239425f9e66_1_en_US_PNG.png) _x000D_ Figure 3-# Search Function - Search by Name_x000D_ (image: 3-imgf1c8ce00a63475219da352391a053f83_1_en_US_PNG.png) _x000D_ Figure 3-# Search Function - Search by Number_x000D_ The search function allows the user to search by the identifier as well._x000D_ (image: 4-imgf0de49e3a6347c1b9da35239129c8368_1_en_US_PNG.png) _x000D_ Figure 3-# Search Function - Search by ID

3.15.3 Online Help

The online documentation consists of released Spectrum Power IMM manuals that have been converted into Portable Document Format (PDF) files. IMM has an integrated online help that provides an extensive guide to the Information Model Management based on the Spectrum Power IMM manuals. Help information can be viewed using the context-sensitive help. This is done by clicking a user interface component and invoking a help viewer to show the related help topic.

3.15.4 National Language Support

National Language Support (NLS) is provided. It is used to customize Spectrum Power IMM with the desired language during configuration time. The character strings are translated using a standard translation software package and the transfer to the system is a system feature.

3.15.5 Main Screen

Once the IMM application starts, it opens the main screen. The main screen represents an application framework for the IMM engineering applications._x000D_ (image: 1-img2fd41e2e69dd902d9da3522a1bd1be21_2_en_US_PNG.png) _x000D_ Figure 3-# IMM Main Screen_x000D_ The main screen is structured into

different areas, dedicated to different aspects of data engineering. It has a standard menu bar at the top of the main window to provide access to some commonly used functions. The top left panel shows the list of datasets available in the IMM database. There is a context menu on the dataset with items tailored to the functions available for the selected dataset. The top right panel shows the list of jobs for the selected dataset. The bottom panel has tabs for the various logging facilities._x000D_ Sections can be resized by dragging the separators between them. The top and bottom halves of the screen can be collapsed by clicking on the title bar for the section. The bottom panel for logging can also be undocked and re-docked.

3.15.6 Job Management

The Job management UI is used for the following:_x000D_ * Create and open a job_x000D_ * Validate and activate a job_x000D_ * Administrate jobs_x000D_ (image: 1-img74e8438a69fcea909da3522a65bc98bb_2_en_US_PNG.png) _x000D_ Figure 3-# Job Management_x000D_ The Job management UI shows the following information about each job that is not finalized:_x000D_ * Job name_x000D_ * Job status_x000D_ * Job owner_x000D_ * Job description_x000D_ Depending on the configured scope of functionality, the following additional information is included: _x000D_ * Job status on PROD (for systems having a separate QAS)_x000D_ * Activation or undone time (in IMM static data model archive only)

3.15.7 Type Editor

The Type Editor is used for the following:_x000D_ * Create new types_x000D_ * View and edit existing types_x000D_ (image: 1-img0dedb980e74f19279da3522a3791b295_1_en_US_PNG.png) _x000D_ Figure 3-# Type Editor_x000D_ There is a read-only mode available called Type Inspector. The type inspector is used to view the existing types._x000D_ _x000D_ (image: 2-Tip.png) | NOTE Types can only be modified and created in the root job of a dataset by users with appropriate access rights. It is recommended to use the Type Editor for small changes only. Modifications of attributes or associations of existing types are not recommended at all. For major changes, external data modeling tools shall be used.Type changes may introduce additional source code changes to take effect in the Spectrum Power 7 system.

3.15.8 Model and Graphics Editor

The Model and Graphics Editor is used to:_x000D_ * View and modify instance properties including links_x000D_ * Create a new instance_x000D_ * View, create, or modify network displays_x000D_ (image: 1-imgb53fe8e774f5ab749da3522a362d834f_2_en_US_PNG.png) _x000D_ Figure 3-# Model and Graphics Editor_x000D_ The screen is structured into different areas, dedicated to different views to the data. It appears once a job is being opened. A set of editors and tools optimized for the various engineering workflows allow data access and definition covering all aspects of data engineering. It has a standard menu bar at the top of the main window to

provide access to some commonly used functions. The left panel shows a tree view of the model data. The attributes panel on the right is able to present various views of the object attributes. Between both panels, multiple schematic diagrams and reports can be opened. The model is edited by dragging new objects into the diagrams and connecting their terminals together, disconnecting objects, deleting objects, or by editing the parameters of the objects in the resizable attributes panel._x000D_ The panels can be resized by dragging the separators between them._x000D_ Diagram Concept_x000D_ A diagram consists of layers in which the presentation details can be distributed to several layers with potentially different magnification or visibility levels._x000D_ The keypoints are as follows:_x000D_ * Support for regular layers_x000D_ A display consists of a set of layers where decluttering by zooming is possible. The layer concept helps the display designer to organize the grouping and visibility of the network component representation._x000D_ * Support for overlay layers_x000D_ Overlays are special layers. They can be toggled on/off in run-time regardless of the zoom level._x000D_ * Layer-based decluttering_x000D_ Each object is assigned to a layer. Layers can be defined to be visible only within a certain zoom range._x000D_ Only those objects are displayed that belong to a currently visible layer._x000D_ Technological Editing_x000D_ Technological editing allows creation, modification, and deletion of electrical power system domain data and topology in a graphical way during diagram construction._x000D_ The following workflows are supported:_x000D_ * Create new graphical objects and new domain data using drag and drop from the topology diagram toolbar._x000D_ * Create new graphical objects and new domain data using copy and paste within and across simultaneous displays._x000D_ * Link graphical objects to existing domain data._x000D_ * Create and manipulate the topology of the electrical power system network._x000D_ * Delete domain data using deletion of graphic objects linking this domain data._x000D_ * Creation, modification and maintenance of symbology._x000D_ Templates_x000D_ The template mechanism available in the IMM UI is extremely powerful. Templates can be used to add pre-packaged sets of data and/or graphics to the database. Depending on the template type, you can contain JavaScript code that allows for scripting the set of data/actions to be performed against the database. Templates can be used to add, delete, and modify/manipulation both domain and graphic data objects._x000D_ Regardless of type, the templates are XML/XDF based. They are persistently stored in IMM’s AC Dataset. When editing/modifying templates, it is recommended that the template be exported from IMM to XDF format, and then edited in a standard XML editor such as XMLSpy or Notepad++. When finished, the template can be loaded back into the IMM database and is immediately available/accessible to all users on the system._x000D_ Some templates can contain advanced logic encoded with embedded JavaScript. This allows you to programmatically navigate through the IMM model and retrieve/manipulate data. This ability to add, delete, and modify content in the database applies to both network/topology data and graphic/presentation data._x000D_ Following are the main types of templates (from simplest to most complex): _x000D_ * User-Defined Templates_x000D_ These templates typically contain hardcoded XML blocks with substitution variables and contain very little if any JavaScript logic. User-defined templates represent repeatable patterns of one or more objects created interactively in the IMM UI by select objects and then saving as a repeatable template. A user-defined template can be as simple as a single equipment or could potentially represent an entire substation._x000D_ * Drop Templates_x000D_ These templates typically contain a mix of XML blocks and JavaScript. A drop template typically represents a small building block – for example, a single equipment with some supplementary objects such as tags, labels, or analogs. The drop templates are typically used to simultaneously create

network/domain and graphic data._x000D_ * Auto-Generation Templates_x000D_ These templates are typically close to pure JavaScript, with only the XML header and footer. These templates are used when the network model already exists, and the user wishes to create a graphic representation of the model based on existing equipment and topological relationships._x000D_ * Manipulation Templates_x000D_ These templates are typically close to pure JavaScript. They are similar to drop templates, except they are used to modify existing data. These are used to “fix up” or “correct” existing domain or diagram data programmatically._x000D_ In general, templates allow for data to be: _x000D_ * Added (All template types)_x000D_ * Modified (Manipulation templates)_x000D_ * Deleted (Manipulation templates)_x000D_ * Retrieved and reported (Manipulation Templates)_x000D_ Following are some examples of template usage: _x000D_ * Simultaneously create an equipment and its graphic representation on an existing diagram (Drop Template)._x000D_ * Capture/save a repeatable object pattern in the model/diagram and reuse to create new objects (User-Defined template)._x000D_ * Change the symbol representation for all breakers of a particular type (Manipulation template)._x000D_ * Automatically add a substation title when auto-generating a diagram (Auto-generation template)._x000D_ Some template types (auto-generation templates, drop templates, and manipulation templates) allow for programmatic navigation through the model hierarchy. This is accomplished by querying/retrieving Parent objects, Child objects, Ancestor and Descendant objects, and Links/Associations._x000D_ Examples of this include: _x000D_ * Finding and looping through all child graphic objects on a diagram that represent a particular equipment type._x000D_ * Finding and looping through all graphic objects on a diagram and assigning graphic properties such as symbol logic and style logic based on the domain object type and selected attributes._x000D_ * Finding and looping through all voltage levels under a substation, and add all child equipment to a new auto-generated diagram._x000D_ There is a well-defined and documented Template API that can be leveraged in the templates that contain embedded JavaScript (auto-generation, drop, and manipulation templates). The functions in the Template API represent the basic building blocks. These can then be extended by the user._x000D_ Following are some of the more commonly used template API functions: _x000D_ * ancestorsOfType – retrieves all ancestors of a particular type for current object._x000D_ * childPaths – retrieves all child objects of a particular type for current object._x000D_ * descendantPaths – retrieves all descendants of a particular type for current object._x000D_ * getValue – retrieves value for specified attribute on current object._x000D_ * setValue – assigns value for specified attribute on current object._x000D_ * linksTo – retrieves all objects to which the current object is linked “to”._x000D_ * log – allows for information to be output to the Model Editor log._x000D_ The Templates are stored persistently in IMM’s AC Dataset and are available to all users. However, when developing new, or modifying existing, templates, they can be loaded interactively (via the Debug dialog) into a user’s specific IMM UI session. Templates loaded interactively override what is in the AC Dataset and are only available to that user for that particular session. The recommendation is to use interactive loading for testing templates under construction. Once finalized, they can be applied to the AC Dataset and made available to others._x000D_ Auto-Generation Templates_x000D_ The Auto-generation templates serve as one component of the diagram auto-generation process._x000D_ The main components are: _x000D_ * The auto-generation templates – each equipment type has its own template with embedded logic/rules governing how an equipment is to be represented on the diagram. This includes creation of labels, analogs, tags, quality codes, and any other supplementary objects that are needed in order to adequately/completely represent an equipment on the diagram._x000D_ * The auto-generation

profiles – each profile functions as a “driver” script for generating a diagram. This is what the user selects when auto-generating a diagram. It creates the diagram, creates the layers, assigns a diagram style, cycles through the voltage levels and adds equipment to the diagram (using the auto-generated templates to assign the correct representation), and then calls the layout engine._x000D_ * The layout engine – the layout engine accepts the set of topologically connected equipment from the IMM UI, leverages its internal logic/rules to calculate an optimal layout based on the selected algorithm, and then returns the layout (including calculated x/y positions for all equipment) to the IMM UI._x000D_ User-Defined Templates_x000D_ * The User-defined templates represent the simplest kind of template. They can be created by interactively selecting one or more objects on the diagram (a repeatable equipment “pattern”) and saving as a template. The resulting template is functional “out of the box”, where little additional adjustment is needed._x000D_ * Although it shares the underlying XML structure with other templates, the User-Defined template typically does not contain the same level of logic/processing._x000D_ * A User-Defined template can create both domain (equipment and topology) and graphic data, or it can be tailored to only create domain data._x000D_ * When creating/using a User-defined template, the process is somewhat similar to a copy/paste operation performed in the IMM UI – however, instead of pasting, the Edit Template feature is used to create a template from the copy buffer content._x000D_ * A User-defined template consists primarily of hardcoded XML block, with embedded substitution variables. The set of input parameters (which are presented for the user to interactively enter) is configurable, and each substitution variable corresponds to an input parameter. The substitution variables can be used to populate attribute values and also link paths._x000D_ * Other than the substitution variables, the remainder of the template is essentially hardcoded, thus allowing for a “cookie cutter” result with minimal user input/intervention. It allows the customer to identify the most common equipment “patterns” that are likely to be found the most often, and to create reusable templates for those._x000D_ * A User-defined template can be used “out of the box”, or it can be modified/extended after it is initially exported/saved._x000D_ * To use a User-defined template, it is selected from the dropdown on the Apply Template dialog. The user is presented with the set of configurable input parameters._x000D_ Drop/Manipulation Templates_x000D_ * The drop templates are accessible through the toolbar buttons above the drawing canvas in the IMM UI._x000D_ * The manipulation templates are typically accessible through the drop-down on the Apply Template dialog._x000D_ * Much of the structure/mechanism is similar for drop/manipulation templates and auto-generation templates. However, the drop/manipulation templates allow for interactive input from the user (similar to User-Defined templates). The input parameters are configurable._x000D_ * The drop template toolbar itself is also configurable, and both the toolbar and the drop/manipulation templates are stored in the AC Dataset._x000D_ * To use a drop template, the user simply selects a button from the toolbar and drags/drops it onto the diagram canvas._x000D_ * The template dialog (similar to that of the User-defined templates) will then be presented. A set of input parameters will need to be entered by the user. Which parameters are actually presented, and how, are controlled by the template.

3.15.9 Multi-Instances Editor

The Multi-Instance Editor (MIE) allows for user-defined query filters on a combination of data instances, attributes and associations. The retrieved objects and attributes can then be edited similarly to what is possible in the IMM UI. Queries can be defined, saved and loaded for reuse._x000D_ (image: 1-img566f3cd3a2ce067f9da3522a5eb0fa89_1_en_US_PNG.png) _x000D_ Figure 3-# Multi-Instances Editor

3.15.10 Symbol Editor

The Symbol Editor is used to edit symbols used on diagrams._x000D_ (image: 1-imgc66eccadd3f4a65b9da3522a390d7737_1_en_US_PNG.png) _x000D_ Figure 3-# Symbol Editor_x000D_ Symbols are combinations of graphic primitives. Symbols play an important role for graphical representation of a domain data instance such as a circuit-breaker for the Spectrum Power runtime user interface depending on its current status.

3.15.11 Color Editor

The Color Editor is used to view, modify, and define color values for color instances._x000D_ (image: 1-img38502002e24bb8b29da3522a0562ed85_1_en_US_PNG.png) _x000D_ Figure 3-# Color Editor

3.15.12 Decision Table Editor

The Decision Table editor is used to maintain diagram decision tables for evaluating the presentation of dynamic display objects in the runtime environment based on their status, quality and other information._x000D_ (image: 1-imgf7d7ced8e26b33ac9da3522a5c1df815_1_en_US_PNG.png) _x000D_ Figure 3-# Decision Table Editor

3.15.13 Shape Style Editor

The Shape Style Editor is used to create reusable shape styles which represent a particular graphic property combination that can be assigned to graphic objects instead of assigning multiple properties one after the other._x000D_ (image: 1-img9eb68944e2e11f3c9da3522a3378903a_1_en_US_PNG.png) _x000D_ Figure 3-# Shape Style Editor

3.15.14 Style Group Editor

Style groups are used to apply dynamic styles to display objects. The Style Group Editor is used to maintain unique styles matching the rules configured in the diagram decision tables by the Decision Table Editor._x000D_ (image: 1-img7d8fa466e2f2fd779da3522a3052b381_1_en_US_PNG.png) _x000D_ Figure 3-# Style Group Editor

3.15.15 Style Logic Editor

The Style Logic Editor is used to maintain style logics for evaluating the style of the presentation of dynamic display objects in the runtime environment based on the combination of a decision table and a style group._x000D_ (image: 1-imgde5fe4d7e312a5f99da3522a14a6d540_1_en_US_PNG.png) _x000D_ Figure 3-# Style Logic Editor

3.15.16 Symbol Group Editor

Symbol groups are used to apply dynamic symbols to display objects. The Symbol Group Editor is used to maintain unique symbols matching the rules configured in the diagram decision tables by the Decision Table Editor._x000D_ (image: 1-imgb32b099be331e4989da3522a073c16be_1_en_US_PNG.png) _x000D_ Figure 3-# Symbol Group Editor

3.15.17 Symbol Logic Editor

The Symbol Logic Editor is used to maintain symbol logics for evaluating the symbol of the presentation of dynamic display objects in the runtime environment based on the combination of a decision table and a symbol group._x000D_ (image: 1-img7a9e6f4de36b70d99da3522a6cd16f51_1_en_US_PNG.png) _x000D_ Figure 3-# Symbol Logic Editor

3.15.18 Text Style Editor

The Text Style Editor is used to create reusable text styles which represent a particular graphic property combination that can be assigned to graphic text objects._x000D_ (image: 1-img4c4b3db7e72b1a4b9da3522a5904cc96_1_en_US_PNG.png) _x000D_ Figure 3-# Text Style Editor

3.15.19 Analog Representation Editor

The Analog Representation Editor is used to view the defined analog representation styles._x000D_ (image: 1-img604fd502e73d3cc19da3522a3d9b8128_1_en_US_PNG.png) _x000D_ Figure 3-# Analog Representation Editor