ERDC/CERL TR-07-46

US Army Corps
of Engineers®
Engineer Research and
Development Center

Fort Future

Facility Composer™ and PACES™ Integration

Development of an XML Interface Based on Industry Foundation Classes

Susan D. Nachtigall and Beth A. Brucker

November 2007

Construction Engineering
Research Laboratory

ERDC/CERL TR-07-46
November 2007

Facility Composer™ and PACES™ Integration

Development of an XML Interface Based on Industry Foundation Classes

Susan D. Nachtigall and Beth A. Brucker

Construction Engineering Research Laboratory
U.S. Army Engineer Research and Development Center
2902 Newmark Drive
Champaign, IL 61822-1076

Final report

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Work Unit 55LG3B

Abstract: The Architectural Engineering and Construction (AEC) industry has made substantial effort over the past several years to create a standard facility modeling format that better enables their different software applications to work together. This emerging standard, known as the Industry Foundation Class (IFC), is being developed by the International Alliance for Interoperability (IAI) and can be found in recent releases of commercial AEC software. With the evolution of this facility modeling standard, it is becoming possible to capture criteria and requirements during planning and design, and to reuse these data during the life cycle of the facility.

The Engineer Research and Development Center, Construction Engineering Research Laboratory is currently developing a set of facility 'architectural' programming tools, called Facility ComposerTM (FC). FC supports the capture and tracking of facility criteria and requirements during planning charrettes, which are key to capturing the stakeholders' requirements in the early phases of project development. As the facility program, criteria, and requirements are chosen, these tools populate the IFC object model. This model can then be used for downstream analyses such as cost, sustainability, and physical security. As the model matures, it can be used for the facility's life cycle.

Contents

Figures

Figures **Page**

Preface

This study was conducted for Headquarters, U.S. Army Corps of Engineers (HQUSACE) under Project "Fort Future"; Work Unit 55LG3B, "Fort Future Facility Composer." Construction Engineering Research Laboratory (CERL) technical monitors were Michael P. Case, Program Manager for Fort Future and Beth Brucker, Project Manager for Facility Composer.

The work was managed and executed by the Engineering Processes Branch (CF-N) of the Facilities Division (CF), CERL. The CERL Principal Investigator was Susan D. Nachtigall. Part of this work was completed by Earth Tech, Englewood, CO under DACA42-01-P-00037, DACA42-02-P-0156, W9132T-04-P-0192, and W9132-05-P-0139. Special recognition is due to Van Woods (CF-N) for his contribution in the development of the original mapping scheme and Jeff Heckel (CF-N) for the coding of the IFC-XML export of Facility Composer. Recognition is also due to Verle Heindselman, Cost Engineering Branch Chief at Louisville District (LRL), Lyle Bonham and Al Frye, technical consultants (ACSIM-AR) for their development of the Army Reserve Cost Model in PACES software. Donald K. Hicks is Chief, CF-N, and L. Michael Golish is Chief, CF. The Installations Technical Director is Martin J. Savoie, CEERD-CV-T. The Deputy Director of CERL is Dr. Kirankumar V. Topudurti. The Director of CERL is Dr. Ilker R. Adiguzel.

CERL is an element of the U.S. Army Engineer Research and Development Center (ERDC), U.S. Army Corps of Engineers. The Commander and Executive Director of ERDC is COL Richard B. Jenkins, and the Director of ERDC is Dr. James R. Houston.

1 Introduction

Background

The Architectural Engineering and Construction (AEC) industry has been making a substantial effort over the past several years to create a standard facility modeling format that better enables their different software applications to work together. This emerging standard, known as the Industry Foundation Class (IFC), is being developed by the International Alliance for Interoperability (IAI) and can be found in recent releases of commercial AEC software. With the evolution of this facility modeling standard, it is becoming possible to capture criteria and requirements during planning and design, and then to reuse these data during the facility's life cycle.

The Engineer Research and Development Center, Construction Engineering Research Laboratory (ERDC-CERL) is developing a set of facility "architectural" programming tools, called Facility Composer (FC), to support the capture and tracking of facility criteria and requirements during planning charrettes. As the facility program, criteria, and requirements are chosen during the charrette process, FC tools populate the IFC object model. The information in the object model can then be used for downstream analyses such as cost, sustainability, and force protection.

These tools, used by Army Centers of Standardization, Installation Planners, and Corps of Engineers Districts, will ensure that standard criteria and requirements are the basis of all Army facility designs. In addition, they will allow criteria updates to be generated more rapidly, provide consistent project documentation (improving credibility with Congress), capture facility requirements for Future Combat Systems, and promote virtual teaming through constant data exchange. The tools will be used during charrettes to capture planning and design decisions for the reliable incorporation of lessons learned, to rapidly generate a programmatic cost estimate using an IFC import into the Parametric Construction Cost Estimating System (PACES), and to produce programming reports for required project documentation (i.e., facility cost, allowable area, and requirement's justification).

Description of Facility Composer

The most important concept of FC is that Army-specific and any other owner/operator-specific and computable criteria and requirements are associated with a growing facility model that continues throughout the life cycle of the facility. Following the concept of Views (currently one-on-one vendor view), as demonstrated by Building Lifecycle Interoperable Software (BLIS), the FC process populates the Industry Foundation Class (IFC) model at certain stages in the early planning and architectural programming process. After an initial library template is developed for a facility type in Requirements Composer, a planner uses Planning Composer to turn the facility template into project-specific information during a planning charrette. When satisfied that all requirements are met for the project, Planning Composer will export an IFC file for use in downstream analysis, such as Layout Composer, Parametric Cost Estimate, and DD 1391 Form project documentation. The process below (Figure 1) outlines the proposed interoperability process for FC and other Commercial Off the Shelf (COTS) and Government Off the Shelf (GOTS)-related applications.

Another important aspect of the process to note is the timing of the IFC file exchange between Planning Composer, Layout Composer, and the Life Cycle Model. The Corps of Engineers, as Project Manager on Military Construction (MILCON) projects, contracts the majority of the design work to private Architectural and Engineering (A-E) firms. The points at which FC exchanges an IFC file are also good break points for working within the MILCON contracting process. The IFC modeling standard should eventually eliminate the need to specify vendor-specific formats or products in contract deliverables. For example, construction document deliverables should no longer need to be in Autodesk's DWG or Bentley's DGN file format. The A-E contract scope would specify IFC compliance only, and the A-E would be free to work in their product of choice.

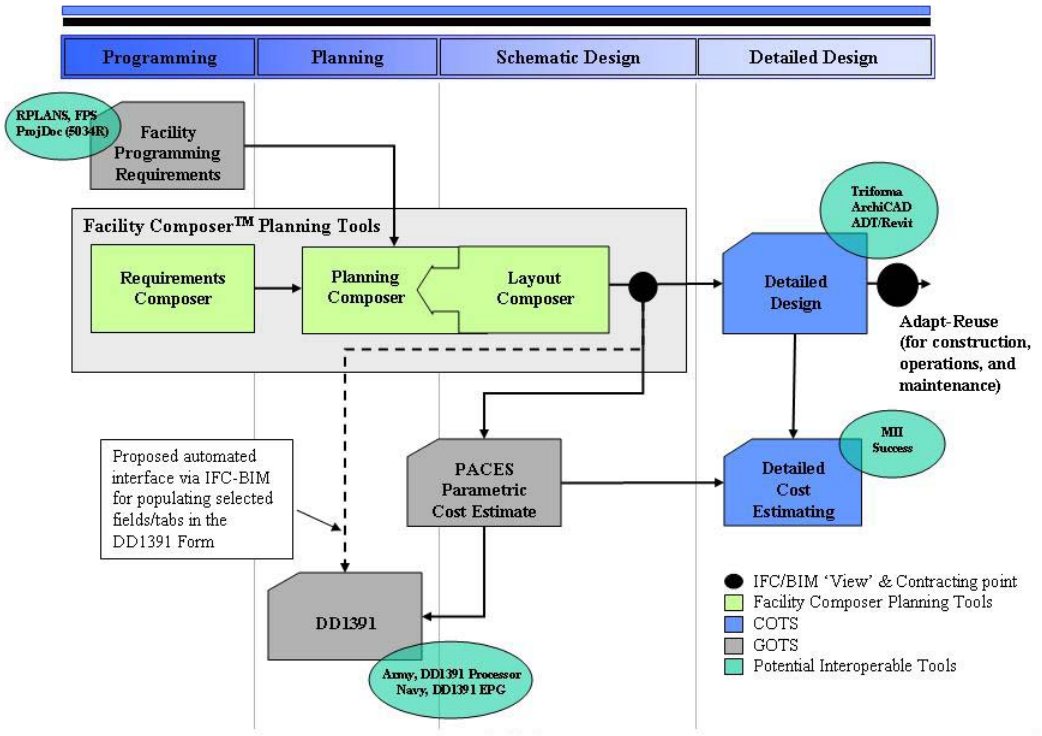


Figure 1. Facility Composer data flow.

In FC, criteria can be associated with different project elements based on the appropriate level of detail, from the **project** to the **site**, the **building**, **story**, **function**, and down to the individual **space**. For example, FC allows one to specify that a target schedule and cost be associated with a project, that masonry exterior walls and a steel structure be used on a building, that 32-watt T-8 fluorescent lights be used in corridors and 50 foot-candles be maintained in the offices, and that a particular room will have vinyl composition tile flooring. FC's ability to maintain a linkage between criteria and project elements (site, building, story, etc.) provides many benefits:

1. It helps in defining criteria and can help in recording their rationale.
2. It helps ensure that critical criteria are followed, and that desired characteristics are recorded and addressed.
3. It helps organize criteria and makes them available at its point of use.
4. It simplifies creation, maintenance, and distribution of new criteria. For example, as requirements that better implement sustainable design principles are developed, these are added to an organization's standard library for use in subsequent projects. These libraries are typically organized around facility type, but this organization is not required.

FC helps support rapid conceptual and detailed design and analysis (cost, structural, heating/ventilating/air conditioning, energy, electrical, operations and maintenance, etc.) either directly or through standards, such as the International Alliance for Interoperability's Industry Foundation Classes (IAI-IFC) and Building Lifecycle Interoperable Software (BLIS).

Owners with numerous facility holdings, in particular, reap benefits from this approach as it helps ensure that the initial design satisfies their corporate criteria, shortening the review process, and avoiding "design by review." All of these benefits result in cost and time savings by reducing user changes late in the design process or during construction. Design quality is also enhanced, as many alternatives can be explored rapidly.

Facility Composer™ Tools

The primary tools in the FC application suite include:

- ***Requirements Composer***, a web-based application that helps in the development of corporate and building-specific criteria libraries
- ***Planning Composer***, which helps users create an architectural program and set values for project-specific criteria
- ***Layout Composer***, which provides an environment for the designer/user to rapidly create two and three dimensional conceptual facility design solutions
- ***Wizards*** that provide support for various discipline specific issues and assist in the completion of individual design tasks and calculations.

The Parametric Cost Engineering System (PACES)

Earth Tech's PACES software is a cost engineering tool to help plan and budget facility and infrastructure construction and renovation costs. PACES is an integrated Windows-based system that prepares parametric cost estimates for new facility construction, renovation, and life-cycle cost. PACES uses pre-engineered model parameters and construction criteria to accurately predict construction costs with limited design information.

This parametric approach differs from traditional cost estimating methods by allowing users to input minimum information to create a cost estimate with model default quantities based on similar projects and experienced A-E assumptions. Predefined and documented engineering relationships link the primary parameters to detailed design assumptions and associated engineering quantities.

In PACES, these quantities can be changed at various places within the model to reflect project-specific conditions. Using parametric models helps avoid errors and omissions that are commonly associated with traditional cost estimating procedures, particularly during planning and early design phases.

Key Benefits

- Construction cost management and evaluation features, including preliminary budget forecasting and project tracking.
- A totally integrated system of A-E parameters, construction criteria, and methodologies. Sophisticated models for general conditions, overhead, and profit quickly calculate indirect and direct costs.
- Proven accuracy to within 7.5 percent of actual construction costs compared to initial budget.
- Complete facility construction and renovation cost estimates with very little up-front engineering data. Modifications can be made during the process to refine the estimates.

Features

- The ability for users to develop detailed estimates based on minimum design information. Those estimates can be applied to any phase of the project.
- Cost estimates are location specific and include general conditions, overhead and profit.
- Assumptions to labor, equipment and material can easily be modified as site conditions change.
- Comprehensive cost database with more than 25,000 line items and location-specific cost adjustments for 2,120 cities worldwide.
- System is updated to reflect current construction and price data.

Objectives

The overall objective of this project was to develop a wizard to demonstrate interoperability between FC and PACES. PACES is a cost estimating system developed by Earth Tech (formerly Talisman Partners) that is used by the U.S. Army Corps of Engineers (USACE) and U.S. Air Force to determine estimates of facility cost from a minimum of detailed information. The assemblies and cost models contained within PACES represent considerable effort on the part of the government. It is highly desirable for tools to reuse, rather than reinvent, these models. This work resulted in

the ability to automatically generate PACES cost estimates based on the space layout, facility criteria, and supporting facilities criteria contained in an FC model.

Approach

To integrate other software applications with PACES, an interface was developed using the self-describing, vendor neutral, eXtensible Markup Language (XML) file format. The PACES import mechanism gathers the information provided in the XML exchange file and supplies any missing information from a set of defaults. With this import capability, other applications would then be able to take advantage of the cost estimating assemblies defined within PACES. The format of the XML exchange file is based on the IAI-IFCs.

Scope

Three different import scenarios were identified for this project. They are outlined in Appendix B. In addition, each scenario had two parts. One method of translation identified was if actual Space objects existed in the FC ifcXML export file and one if they did not. Chapter 3 details how the mappings were determined and addressed. In addition, site supporting requirements were also addressed in order to ensure a more accurate final cost estimate. Appendix A includes the final translation files used by PACES for the import process.

Mode of Technology Transfer

The parametric cost estimating system for USACE is PACES. The goal is to have Earth Tech and all of AEC industry using one open standard for interoperability. By doing this, future versions of PACES will always have the capability to read in FC data to generate a cost estimate.

Since no cost data is contained within FC itself, it is key for the cost models in PACES to remain current. This activity is currently undertaken by Huntsville Center's Cost Engineering Branch.

In addition, Huntsville Center's WebPAX branch has added the capability to upload PACES cost estimates into the DD1391 processor. With this capability, the goal is to get a PACES estimate that would be more accurate than the PC-Cost estimate, especially if FC was used in the process.

2 Process for Developing Interoperability

To integrate other software applications with PACES, an interface was developed using the self-describing, vendor neutral, XML file format. The PACES import mechanism will gather the information provided in the XML exchange file and supply any missing information from a set of defaults. From this standard format, other applications will then be able to take advantage of the cost estimating assemblies defined within PACES.

Methodology

Develop initial mappings

When this work was undertaken, very little documentation existed on how to map the IFC's in an XML format. To ensure that the right approach was being taken initially, Thomas Liebich, the IAI's Modelling Support Group Lead, was contacted on how best to approach this issue. Once guidance was given, mapping the FC data to the IFC format was successful.

Through the interface that was developed in FC, the user is able to populate the IFC object model as they enter information during the planning process. Figures 2 through 7 are examples of how FC populates the IFC model without the user having any knowledge of the IFC object model definitions.

Figure 2 depicts how FC maps specific information into the IFC model. All of the objects are linked together by a relationship entity that exists in the model. For example, *IfcSite* is related to *IfcProject* through *IfcRelAggregates*, which is used to create a hierarchical structure within the model. This linkage also occurs in relating the space to the building story, the building story to the building, and the building to the site.

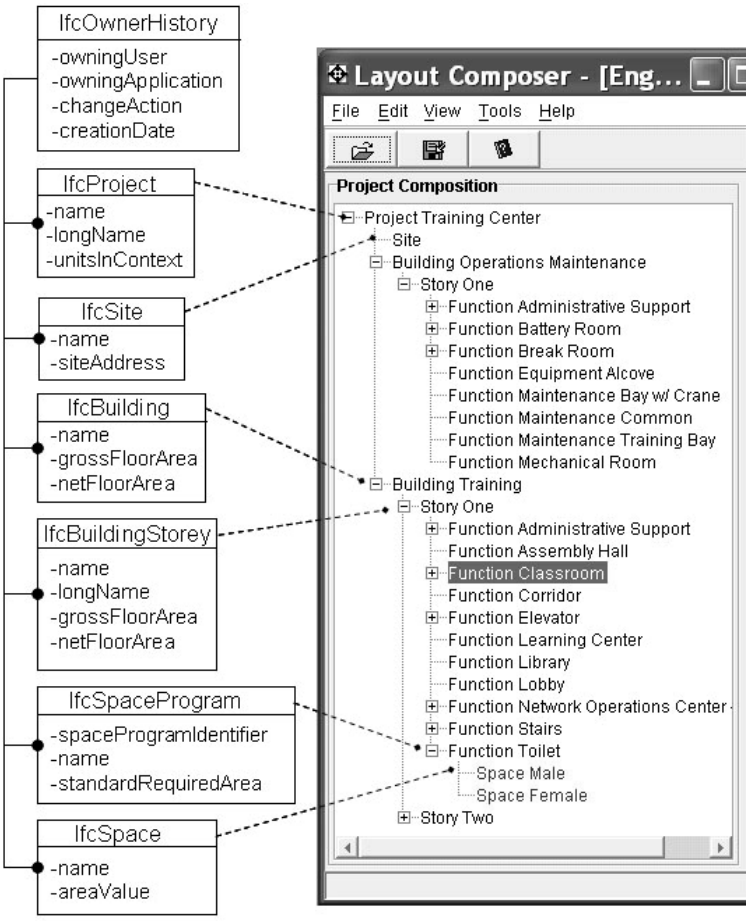


Figure 2. Facility Composer interface for project hierarchy with associated IFC objects.

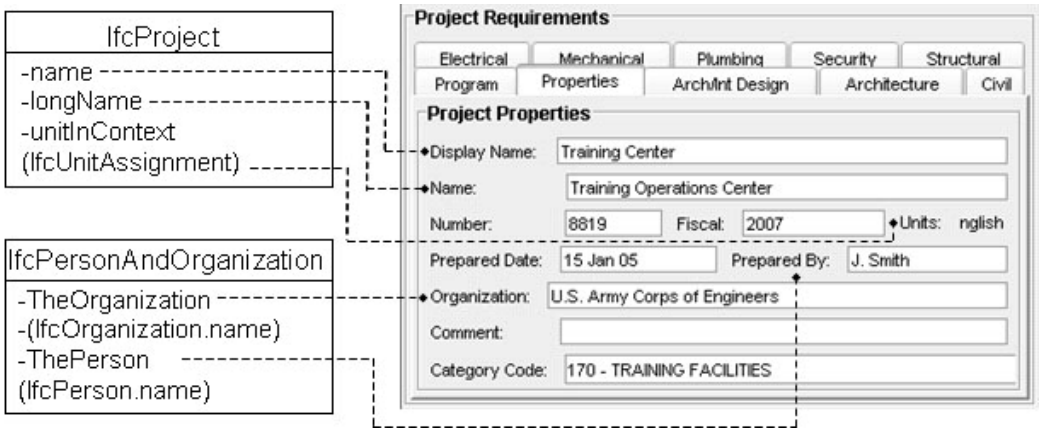


Figure 3. Facility Composer project properties with associated IFC objects.

Similarly, the attributes of *IfcSite* and *IfcPostalAddress* objects are contained in the Properties Tab of the Site Level (Figure 4).

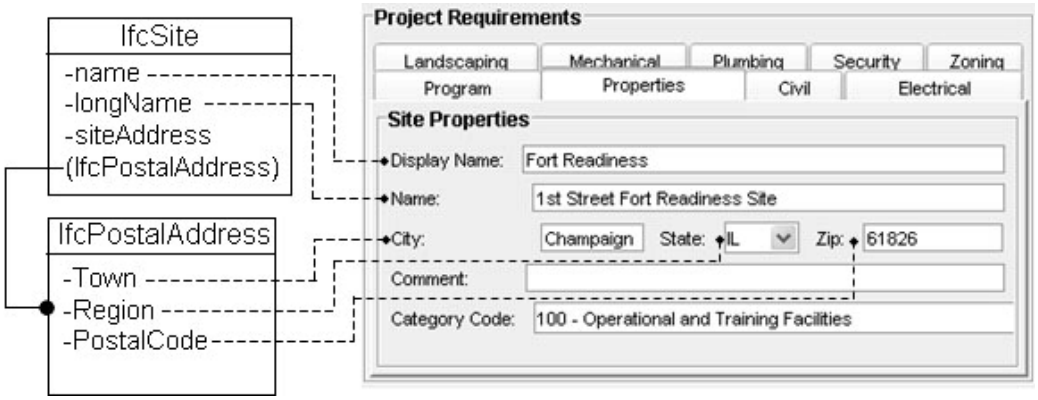


Figure 4. Facility Composer site properties with associated IFC objects.

In addition to simple project information, FC organizes the architectural program in such a way that Planned and Actual spatial requirements are populated. Figure 5 depicts a simple example of the Program Tab in FC at the Function level. The Planned Area in the Building row is exported to the IFC Property Set *Pset_BuildingCommon*, which contains an attribute of *grossPlannedArea*. The property set is defined by the *IfcProperySet* object and attached to the building object through *IfcRelDefinesByProperties*.

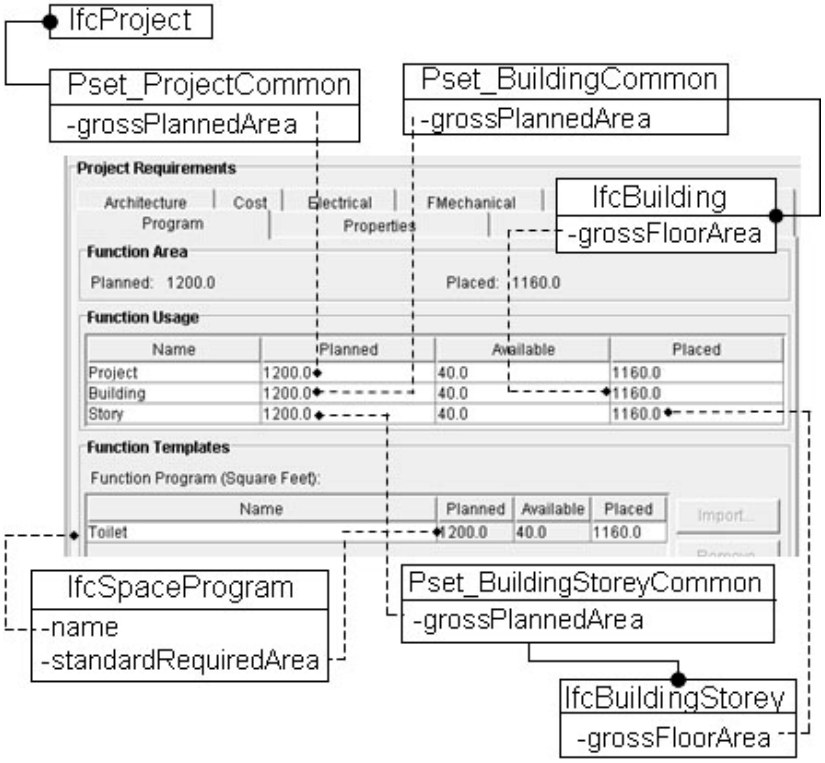


Figure 5. Facility Composer area programming interface with associated IFC objects.

The total Actual Placed area (if spaces have been placed in Layout Composer) is exported to the *IfcBuilding* object. This information is exported

by means of the *IfcElementQuantity* class and is also related to the building through *IfcRelDefinesByProperties*. The planning and actual spatial information at all levels of the project are exported this same way. However, for the Functions themselves, the planned area is exported to the *standardRequiredArea* attribute of the *IfcSpaceProgram* object. Actual areas, placed using Layout Composer, are exported per *IfcSpace* object as seen in Figure 6.

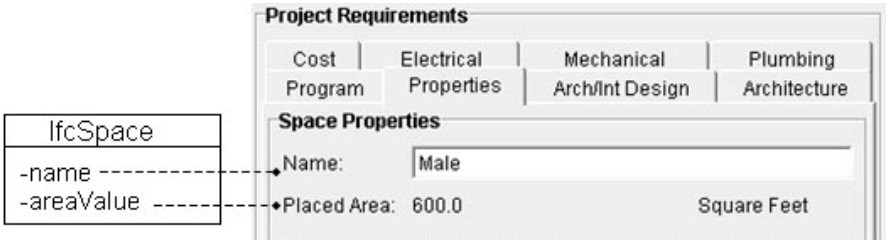


Figure 6. Facility Composer space properties with associated IFC object.

For a criterion that does not have a logical IFC object or attribute, FC was designed to capture these requirements as dynamically extendable properties during the export process. This type of Property Set defines properties for which an entity does not already exist within the IFC model. For example, Figure 7 shows some typical criteria located on the Cost tab within FC. Highlighted within the Requirement column is a requirement called “Cost Models.” As FC exports the information into the IFC model, this requirement will become an *IfcPropertySet* called “Cost Models,” which has properties of “Model Code” and “Model Group.”

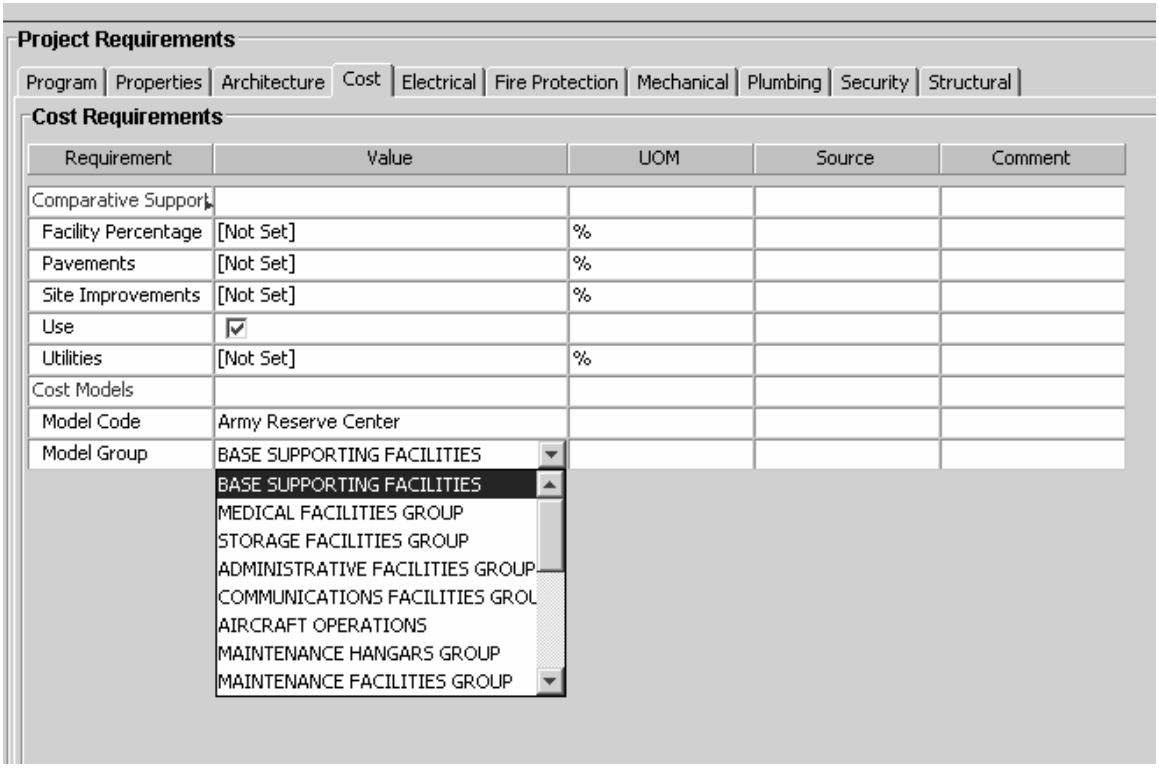


Figure 7. Facility Composer cost tab showing Model Code and Model Group criteria.

Once FC was capable of exporting to the ifcXML format, the next step was to determine what type of data PACES needed to do a cost estimate and make sure the data were available in FC. To accomplish this step, the U.S. Army Engineer Research and Development Center's Construction Engineering Research Laboratory (ERDC-CERL) contracted with Earth Tech, at that time know as Talisman, to request as the first deliverable a document that would outline the types of data that the software requires (see Appendix B). This document outlined the data requirements for generating a cost estimate within their software.

Once this step was completed, ERDC-CERL began the development of a mapping schema between FC and the PACES data model. To meet this need, ERDC-CERL developed a Mapping Spreadsheet to start identifying the first round of mappings for data between FC and PACES was based originally on IFC 2.0 (see Appendix D). These initial mappings were the basis of all future work that was conducted between ERDC-CERL and Earth Tech.

After a first draft was developed, ERDC-CERL and Earth Tech began the task of testing sample files. Through this, many modifications needed to be made to the schema and the FC data model itself.

In addition, while ERDC-CERL developed the first draft schema, Earth Tech began the process of modifying PACES so that it can read XML files conforming to the interface specification defined. The resulting interface developed by Earth Tech is at Appendix C.

Revise mappings

After the completion of the first phase of mappings, the IFC model on which the export was based went through a major change. The IAI released a new version of the IFCs (2.x), which changed the core model (the kernel). The changes in the new version caused the need for ERDC-CERL to update the entire object model of FC itself. After the update, the mapping developed in the first phase of development would no longer be useful. To resolve this problem, ERDC-CERL contracted with Earth Tech once again to update their import code based on this new mapping schema.

In addition, Earth Tech was required to update and refine the graphical user interface that was developed during the previous phase. The items that needed to be addressed and updated based on feedback were: general sequence and approach, clear understanding of interface in terms of identifying (1) default assigned values and (2) missing values that require user input, and finally the clarity of any applicable error messages.

Site work criteria

Once the overall interoperability of basic facility data was established between FC and PACES, it was determined that the exchanging of site requirements would further ensure that the cost estimate being uploaded to the 1391 processor would be more accurate. To accomplish this exchange, another contract was executed between ERDC-CERL and Earth Tech to address the capability of importing Supporting Facility. Since PACES contains several supporting facility models, it was decided to concentrate on only four of them in order to demonstrate the capability. The supporting facilities selected were Parking Lots, Underground Electrical, Fencing, and Water Distribution.

PACES data export

The information within PACES itself is not static. So it was determined that the only way to keep the associated FC criteria synchronized was with the creation of a tool to allow users to update the data within FC with the current data in PACES.

The second task was to create a utility to generate an XML file from the PACES database to provide updated *Model* and *Functional Space Area* for use in the FC model to assist in the mapping of Functional Space Area types to FC functions. This means that a small utility was developed to extract Functional Space Area codes and descriptions as well as other possible data from the PACES.mdb file and input to an XML format. It was anticipated that, while the data structures described in the PACES Interface Specification Document (Appendix B) are stable, the information they contain will undergo continual modification and improvement.

Army Reserve

In 2003 the Army Reserve, long-time customers of ERDC-CERL, began the transition process from the Modular Design System to FC for their planning tool. For the transition to be successful, the Army Reserve needed to have cost models for their facility types added to the PACES database.

During the testing of the PACES import with regard to the Army Reserve testing plan, an issue was found with the way in which the PACES import process searched the FC ifcXML file for Functional Space Area names. When the PACES XML Import algorithm imports a FSA element, it attempts to locate a suitable FSA in the PACES FSA. TableinValue represents the FSA Name from the XML file.

1. Searching FSAs under the Current Facility Model
   a. Find First Record where [PACES CODE] = inValue
   b. Find First Record where [PACES FSA DESC] = inValue
   c. Find First Record where [PACES FSA DESC] Like inValue*
   d. If the First 3 Characters are Upper-Case and Fourth Character is White-Space, then Find First Record where [PACES CODE] = First 3 Characters of inValue
   e. Replace all Non-Alpha Characters in inValue with a * and perform a Like Match with [PACES FSA DESC]
   f. If the First 3 Characters are Upper-Case and Fourth Character is White-Space, then Replace all Non-Alpha Characters After the first three characters of inValue with a * and perform a Like Match with [PACES FSA DESC]
2. Searching All FSAs regardless of Current Facility Model
   a. Find First Record where [PACES CODE] = inValue
   b. Find First Record where [PACES FSA DESC] = inValue
   c. Find First Record where [PACES FSA DESC] Like inValue*

d. If the First 3 Characters are Upper-Case and Fourth Character is White-Space, then Find First Record where [PACES CODE] = First 3 Characters of inValue
e. Replace all Non-Alpha Characters in inValue with a * and perform a Like Match with [PACES FSA DESC]
f. If the First 3 Characters are Upper-Case and Fourth Character is White-Space, then Replace all Non-Alpha Characters After the first three characters of inValue with a * and perform a Like Match with [PACES FSA DESC]

If there is a successful hit under 1, it never tries number 2. So if Stairwells are not defaulted in the model you are working with, but another FSA that would match is, it will end up using that FSA.

As an example, with "MDS STR" as the inValue you would see:

1.a : No Match - Since "MDS STR" is not a PACES CODE
1.b : No Match - Since there is no FSA Description Explicitly called "MDS STR"
1.c : No Match - Since "MDS STR STAIRWELL" is not a default FSA for this model
1.d: No Match - Since none of the PACES Codes are "MDS"
1.e : You have a Match Against "MDS ADC ADMINISTRATIVE COMMON" - Since this is the first record that satisfies the Search "*MDS*STR*" or Contains MDS followed by STR anywhere in the Description Name

2.a : No Match - Since "MDS STR" is not a PACES CODE
2.b : No Match - Since there is no FSA Description Explicitly called "MDS STR"
2.c : Match - Since "MDS STR STAIRWELL" Begins with "MDS STR"

Since there was a positive match under 1, the translator will favor the model matches over the non-model matches in 2.

To resolve this issue, a new requirement in FC was created called “Function Model.” It is here under the “Code” property that the Function/Space is mapped to the actual FSA code in PACES. This mapping ensures that the correct FSA is used.

Through the Reserve transition, another import capability was requested. The Army Reserve wanted to be able to use the PACES concept of Comparative Supporting Facilities to develop their cost estimates. To meet this need, the capability to translate this information within FC and PACES was needed.

Final results

The results of all the above work were successful. The following chapters present the final product. Chapter 3 outlines where the PACES import algorithm will find the required data within the Facility Composer ifcXML file based on the original Scope of Work outlined in Appendix B. Chapter 4 addresses the mapping location of supporting facilities criteria and finally Chapter 5 describes the location of the information for the Comparative Supporting Facilities parameters.

In developing the mappings for the integration, this project has resulted in the development within the IFCs of a new view definition that addresses the ability to transfer Early Planning and Design information to the Parametric Cost Model and detailed design. This continued work is to be added to the IAI's list of view definitions and will also be included in the first release of the National Building Information Modelling Standard.

3 Final Mapping Information

PACES Project Data Fields

If any of the below information is missing, the translator will ask the user to supply the information during the PACES import process.

Project Name

```
<Project id="i108">
    <globalId>CIfcproject_x0x_-179154745639284385 9</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Raleigh Durham</name>
    <description>this is a test for export</description>
      <longName/>
```

Description

```
<Project id="i108">
    <globalId>CIfcproject_x0x_-179154745639284385 9</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Raleigh Durham</name>
    <description>this is a test for export</description>
      <longName/>
```

State

```
<Site id="i621">
    <globalId>CIfcsite_x7x_-1717185672987352433</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Raleigh</name>
    <description/>
    <longName/>
    <compositionType>element</compositionType>
    <siteAddress>
        <PostalAddress>
            <town>Raleigh</town>
            <region>NC</region>
            <postalCode>36524</postalCode>
        </PostalAddress>
    </siteAddress>
</Site>
```

City

```
<Site id="i621">
    <globalId>CIfcsite_x7x_-1717185672987352433</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Raleigh</name>
    <description/>
    <longName/>
    <compositionType>element</compositionType>
    <siteAddress>
        <PostalAddress>
            <town>Raleigh</town>
            <region>NC</region>
            <postalCode>36524</postalCode>
        </PostalAddress>
    </siteAddress>
</Site>
```

Location Code

Not being exported at this time. Simply use the State and City information to relate to the best match of the PACES data.

Units Option

The unitsInContext information is associated to the Project object within the ifcXML file.

```
<unitsInContext>
    <UnitAssignment>
        <units>
            <ConversionBasedUnit>
                <dimensions>
                    <DimensionalExponents>
                        <lengthExponent>0</lengthExponent>
                        <massExponent>0</massExponent>
                        <timeExponent>0</timeExponent>
                        <electricCurrentExponent>0</electricCurrentExponent>
                        <thermodynamicTemperatureExponent>1</thermodynamicTemperatureExponent>
                        <amountOfSubstanceExponent>0</amountOfSubstanceExponent>
                        <luminousIntensityExponent>0</luminousIntensityExponent>
                    </DimensionalExponents>
                </dimensions>
                <unitType>thermodynamictemperatureunit</unitType>
                <name>deg f</name>
                <conversionFactor>
                    <MeasureWithUnit>
                        <valueComponent>
                            <Real>-17.22222</Real>
                        </valueComponent>
                        <unitComponent>
                            <SIUnit href="i9" xsi:nil="true"/>
                        </unitComponent>
                    </MeasureWithUnit>
                </conversionFactor>
            </ConversionBasedUnit>
            <ConversionBasedUnit>
                <dimensions>
                    <DimensionalExponents>
                        <lengthExponent>2</lengthExponent>
```

```
                    <massExponent>0</massExponent>
                    <timeExponent>0</timeExponent>
                    <electricCurrentExponent>0</electricCurrentExponent>
                    <thermodynamicTemperatureExponent>0</thermodynamicTemperatureExponent>
                    <amountOfSubstanceExponent>0</amountOfSubstanceExponent>
                    <luminousIntensityExponent>0</luminousIntensityExponent>
                </DimensionalExponents>
            </dimensions>
            <unitType>areaunit</unitType>
            <name>sf</name>
            <conversionFactor>
                <MeasureWithUnit>
                    <valueComponent>
                        <Real>0.092903</Real>
                    </valueComponent>
                    <unitComponent>
                        <SIUnit href="i8" xsi:nil="true"/>
                    </unitComponent>
                </MeasureWithUnit>
            </conversionFactor>
        </ConversionBasedUnit>
        <ConversionBasedUnit>
            <dimensions>
                <DimensionalExponents>
                <lengthExponent>1</lengthExponent>
                    <massExponent>0</massExponent>
                    <timeExponent>0</timeExponent>
                    <electricCurrentExponent>0</electricCurrentExponent>
                    <thermodynamicTemperatureExponent>0</thermodynamicTemperatureExponent>
                    <amountOfSubstanceExponent>0</amountOfSubstanceExponent>
                    <luminousIntensityExponent>0</luminousIntensityExponent>
                </DimensionalExponents>
            </dimensions>
            <unitType>lengthunit</unitType>
            <name>inch</name>
            <conversionFactor>
                <MeasureWithUnit>
                    <valueComponent>
                        <Real>25.4</Real>
                    </valueComponent>
                    <unitComponent>
                        <SIUnit href="i7" xsi:nil="true"/>
                    </unitComponent>
                </MeasureWithUnit>
            </conversionFactor>
        </ConversionBasedUnit>
        <ConversionBasedUnit>
            <dimensions>
                <DimensionalExponents>
                    <lengthExponent>3</lengthExponent>
                    <massExponent>0</massExponent>
                    <timeExponent>0</timeExponent>
                    <electricCurrentExponent>0</electricCurrentExponent>
                    <thermodynamicTemperatureExponent>0</thermodynamicTemperatureExponent>
                    <amountOfSubstanceExponent>0</amountOfSubstanceExponent>
                    <luminousIntensityExponent>0</luminousIntensityExponent>
                </DimensionalExponents>
            </dimensions>
            <unitType>volumeunit</unitType>
            <name>cy</name>
            <conversionFactor>
                <MeasureWithUnit>
                    <valueComponent>
                        <Real>0.7645549</Real>
                    </valueComponent>
                    <unitComponent>
                        <SIUnit href="i6" xsi:nil="true"/>
                    </unitComponent>
                </MeasureWithUnit>
            </conversionFactor>
        </ConversionBasedUnit>
    </units>
```

```
    </UnitAssignment>
</unitsInContext>
```

Service

```
    <Organization id="i2">
        <name>Army</name>
    </Organization>
```

Project Number

Linked to the Project Object

```
<PropertySet id="i109">
    <globalId>PropertySet_xi109x_8557687608271169895</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Pset_CCProjectData</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Number</name>
            <nominalValue>
                <Integer>294857</Integer>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i110">
    <globalId>RelDefinesByProperties_xi110x_8557687608271169895</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Project href="i108" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i109" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Prepared By

```
<Person id="i1">
    <familyName>s-nachtigall</familyName>
</Person>
```

Prepared Date

Two different methods could be used to find this particular information. The preferred method would be to simply use the timestamp of the ifcXML file creation. To use the timestamp method, the information would be located within the OwnerHistory section of the xml file.

```
    <OwnerHistory id="i4">
        <owningUser>
            <PersonAndOrganization href="i3" xsi:nil="true"/>
        </owningUser>
        <owningApplication>
```

```
            <Application>
                <applicationDeveloper>
                    <Organization href="i2" xsi:nil="true"/>
                </applicationDeveloper>
                <version>1.0</version>
                <applicationFullName>BCCore to IFC 2X</applicationFullName>
                <applicationIdentifier>BCCore</applicationIdentifier>
            </Application>
        </owningApplication>
        <changeAction>added</changeAction>
        <creationDate>20041109</creationDate>
    </OwnerHistory>
```

The second method would be to use Planning Composer specific criteria. The only negative to this method is that, if one would try to import a compliant ifcXML file generated from another program, the needed information would not be there. The prepared date in this case would be in the following location, which is linked to the Project object.

```
<PropertySet id="i109">
    <globalId>PropertySet_xi109x_8557687608271169895</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Pset_CCProjectData</name>
    <hasProperties>
        <PropertySingleValue>
            <name>PreparedDate</name>
            <nominalValue>
                <Text>11/09/04</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i110">
    <globalId>RelDefinesByProperties_xi110x_8557687608271169895</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Project href="i108" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i109" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Program Year

Linked to the Project object

```
<PropertySet id="i109">


    <globalId>PropertySet_xi109x_8557687608271169895</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Pset_CCProjectData</name>
    <hasProperties>
        <PropertySingleValue>
```

```
            <name>Fiscal</name>
            <nominalValue>
                <Integer>2008</Integer>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i110">
    <globalId>RelDefinesByProperties_xi110x_8557687608271169895</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Project href="i108" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i109" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Project Comments

```
<Project id="i108">
    <globalId>CIfcproject_x0x_-1791547456392843859</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Raleigh Durham</name>
    <description>this is a test for export</description>
    <longName/>
```

PACES Facility Data Fields

Facility Name

```
<Building id="i672">
    <globalId>CIfcbuilding_x18331x_1581238087589554042</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>I. 171 - TRAINING BUILDING</name>
    <description/>
    <longName/>
    <compositionType>element</compositionType>
</Building>
```

Facility Description

```
<Building id="i672">
    <globalId>CIfcbuilding_x18331x_1581238087589554042</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>I. 171 - TRAINING BUILDING</name>
    <description/>
    <longName/>
    <compositionType>element</compositionType>
</Building>
```

Facility Comments

```
<Building id="i672">
    <globalId>CIfcbuilding_x18331x_1581238087589554042</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>I. 171 - TRAINING BUILDING</name>
    <description/>
    <longName/>
    <compositionType>element</compositionType>
</Building>
```

PACES Model Category

Related to Building Object

```
<!-- ===== MDSREQUIREMENTSET : Cost Models ===== -->
<PropertySet id="i707">
    <globalId>CIfcroot_x1305x_-4049112392319854768</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Cost Models</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Model Code</name>
            <nominalValue>
                <Text>Army Reserve Center</Text>
            </nominalValue>
        </PropertySingleValue>
        <PropertySingleValue>
            <name>Model Group</name>
            <nominalValue>
                <Text>Administrative Facilities Group</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i708">
    <globalId>RelDefinesByProperties_xi708x_-4049112392319854768</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Building href="i672" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i707" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

PACES Model Code

Related to Building Object

```
<!-- ===== MDSREQUIREMENTSET : Cost Models ===== -->
<PropertySet id="i707">
    <globalId>CIfcroot_x1305x_-4049112392319854768</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
```

```
    </ownerHistory>
    <name>Cost Models</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Model Code</name>
            <nominalValue>
                <Text>Army Reserve Center</Text>
            </nominalValue>
        </PropertySingleValue>
        <PropertySingleValue>
            <name>Model Group</name>
            <nominalValue>
                <Text>Administrative Facilities Group</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i708">
    <globalId>RelDefinesByProperties_xi708x_-4049112392319854768</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Building href="i672" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i707" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Gross Floor Area

As described in Chapter 2, when actual IfcSpace object exists, the translator should total up the ElementQuantity.QuantityArea values for each space.

```
<!-- ================================================================ -->
<!-- ================================================================ -->
<!-- ===== CIFCSPACE ===== -->
<Space id="i321">
    <globalId>CIfcspace_x20355x_-4000794154198605858</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Armory</name>
    <representation>
        <ProductDefinitionShape href="i324" xsi:nil="true"/>
    </representation>
    <compositionType>partial</compositionType>
    <interiorOrExteriorSpace>notdefined</interiorOrExteriorSpace>
</Space>
<!--================================================================ -->
<!-- ===== BoundingBox ===== -->
<BoundingBox id="i322">
    <corner>
        <CartesianPoint>
            <coordinates>-768.0 -31.999999999999996 0.0</coordinates>
        </CartesianPoint>
    </corner>
    <xDim>-1.0</xDim>
    <yDim>0.0</yDim>
    <zDim>0.0</zDim>
</BoundingBox>
<ShapeRepresentation id="i323">
```

```
        <contextOfItems>
            <GeometricRepresentationContext href="i5" xsi:nil="true"/>
        </contextOfItems>
        <representationType>BoundingBox</representationType>
        <items>
            <BoundingBox href="i322" xsi:nil="true"/>
        </items>
    </ShapeRepresentation>
    <ProductDefinitionShape id="i324">
        <representations>
            <ShapeRepresentation href="i323" xsi:nil="true"/>
        </representations>
    </ProductDefinitionShape>
    <ElementQuantity id="i325">
        <globalId>ElementQuantity_xi325x_-39359100010662773 52</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <methodOfMeasurement>unknown</methodOfMeasurement>
        <quantities>
            <QuantityArea>
                <name>Area</name>
                <areaValue>1024.0</areaValue>
            </QuantityArea>
        </quantities>
    </ElementQuantity>
```

If actual spaces do not exist, the total PlannedArea of the building should be used. This value can be found in the following location:

```
<name>Pset_CCBuildingData</name>
    <hasProperties>
        <PropertySingleValue>
            <name>GrossPlannedArea</name>
            <nominalValue>
                <Real>302664.0</Real>
            </nominalValue>
        </PropertySingleValue>
```

Stories Above Grade

To calculate the number of stories above grade, the translator will need to determine the number of Storey objects which have an elevation above 0. A storey within the ifcXML file is identified as *BuildingStorey*. Its elevation information can be found in the following location within the *BuildingStorey* object.

```
<BuildingStorey id="i992">
    <globalId>CIfcbuildingstorey_x8833x_6899518654599681342</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>2</name>
    <description/>
    <longName/>
    <compositionType>element</compositionType>
    <elevation>14.0</elevation>
</BuildingStorey>
```

If there are no *BuildingStorey* objects within the ifcXML file, simply use the PACES default values for the specific Model Group and Model Code.

Stories Below Grade

This process is the same as calculating stories above grade; however, the storey objects should have an elevation below 0.

PACES Facility Quantity Parameters

Building Footprint

Use existing PACES algorithms to calculate this criterion based on the Gross Floor Area (GFA) in XML file.

Building Perimeter

Use existing PACES algorithms to calculate this criterion based on the GFA in XML file.

Roof Area

Use existing PACES algorithms to calculate this criterion based on the GFA in XML file.

Facility Floor to Floor Height Above Grade

The determination of whether a floor is above or below grade is based on the elevation of that particular storey. An example of the XML is as follows.

```
<!-- ===== MDSBUILDINGSTOREY ===== -->
    <BuildingStorey id="i1219">
        <globalId>CIfcbuildingstorey_x18566x_26197075106519378</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>1</name>
        <description/>
        <longName/>
        <compositionType>element</compositionType>
        <elevation>0.0</elevation>
    </BuildingStorey>
```

In the XML above, the floor described in the dataset would be considered above grade. After the elevation is determined, the actual floor to floor height can be found within a PropertySet linked to the Storey object. Here is an example:

```
<!-- ===== MDSREQUIREMENTSET : BCSystem ===== -->
   <PropertySet id="i699">
      <globalId>CIfcroot_x35019x_-8845784696554510022</globalId>
      <ownerHistory>
         <OwnerHistory href="i4" xsi:nil="true"/>
      </ownerHistory>
      <name>BCSystem</name>
      <hasProperties>
         <PropertySingleValue>
            <name>FloorToFloorHeight</name>
            <nominalValue>
               <Real>14.0</Real>
            </nominalValue>
            <unit>
               <ConversionBasedUnit>
                  <dimensions>
                     <DimensionalExponents>
                        <lengthExponent>1</lengthExponent>
                        <massExponent>0</massExponent>
                        <timeExponent>0</timeExponent>
                        <electricCurrentExponent>0</electricCurrentExponent>
                        <thermodynamicTemperatureExponent>0</thermodynamicTemperatureExponent>
                        <amountOfSubstanceExponent>0</amountOfSubstanceExponent>
                        <luminousIntensityExponent>0</luminousIntensityExponent>
                     </DimensionalExponents>
                  </dimensions>
                  <unitType>lengthunit</unitType>
                  <name>ft</name>
                  <conversionFactor>
                     <MeasureWithUnit>
                        <valueComponent>
                           <Real>304.8</Real>
                        </valueComponent>
                        <unitComponent>
                           <SIUnit href="i7" xsi:nil="true"/>
                        </unitComponent>
                     </MeasureWithUnit>
                  </conversionFactor>
               </ConversionBasedUnit>
            </unit>
         </PropertySingleValue>
      </hasProperties>
   </PropertySet>
```

If no *BuildingStorey* objects are within the ifcXML file, simply use the PACES default values for the specific Model Group and Model Code.

Facility Floor to Floor Height Below Grade

This parameter is determined the same way as the criteria just described, although the elevation would need to be below grade/sea level, a negative number.

Exterior Wall Area

The translator should use the existing PACES Model Code/Group algorithms based on the GFA in the XML file.

Exterior Window Area

The translator should use the existing PACES Model Code/Group algorithms based on the GFA in the XML file.

Facility Exterior Doors

The criteria can be found in a *PropertySet* related to the Building object. If the *PropertySingleValue* for the criteria is 0, simply use the PACES defaults for the associated cost model.

```
<PropertySet id="i727">
    <globalId>CIfcroot_x35033x_-886891947447298970 2</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>PACES</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Exterior Doors</name>
            <nominalValue>
                <Text>0</Text>
            </nominalValue>
```

Facility Special Doors

The translator should use the values in the existing PACES Model Code/Group algorithms based on the GFA in the XML file.

Facility Floor to Ceiling Height Above Grade

The determination of whether a floor is above or below grade is based on the elevation of that particular storey. An example of this in XML is as follows.

```
<!-- ===== MDSBUILDINGSTOREY ===== -->
    <BuildingStorey id="i1219">
        <globalId>CIfcbuildingstorey_x18566x_2619707510651937 8</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>1</name>
        <description/>
        <longName/>
        <compositionType>element</compositionType>
        <elevation>0.0</elevation>
    </BuildingStorey>
```

In the case shown above, the floor described below would be considered above grade. After the elevation is determined, the actual floor to floor height can be found within a PropertySet linked to the Storey object. Here is an example:

```
<PropertySet id="i1286">
        <globalId>CIfcroot_x55699x_-7712884446782664482</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>BCSystem</name>
        <hasProperties>
            <PropertySingleValue>
                <name>FloorToCeilingHeight</name>
                <nominalValue>
                    <Real>8.67</Real>
                </nominalValue>
```

If no *BuildingStorey* objects are within the ifcXML file, simply use the PACES default values for the specific Model Group and Model Code.

Facility Floor to Ceiling Height Below Grade

This parameter is determined the same way as the criteria just described above, although the elevation would need to be below grade/sea level, a negative number.

Facility Stairwells

To determine the number of stairwells in the export file, the translator needs to could the number of stair spaces within the file. If no actual spaces are placed, default values should be used. Keep in mind that, as described in Chapter 1, the actual FSA Codes are being exported.

Facility Elevators

This process is the same as determining the number of stairwells.

Facility Plumbing Domestic Water

Use the PACES default values based on the Model Group and Model Code in the XML file.

Facility Plumbing Sanitary Waste

Use the PACES default values based on the Model Group and Model Code in the XML file.

Facility Plumbing Special Systems

Use the PACES default values based on the Model Group and Model Code in the XML file.

Facility Plumbing

Use the PACES default values based on the Model Group and Model Code in the XML file.

Facility Electrical Load

The criteria can be found in a *PropertySet* named **Electrical Loads** related to the Building object. If the *PropertySingleValue* for the criteria is 0, use the PACES defaults for the associated cost model.

```
<PropertySet id="i447">
 <globalId>CIfcroot_x28501x_3212906881209983532</globalId>
 <ownerHistory>
      <OwnerHistory href="i4" xsi:nil="true"/>
 </ownerHistory>
 <name>Loads</name>
 <hasProperties>
      <PropertySingleValue>
           <name>Electrical Loads</name>
           <nominalValue>
                <Text>100</Text>
           </nominalValue>
      </PropertySingleValue>
 </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i448">
 <globalId>RelDefinesByProperties_xi448x_3212906881209983532</globalId>
 <ownerHistory>
      <OwnerHistory href="i4" xsi:nil="true"/>
 </ownerHistory>
 <relatedObjects>
      <Building href="i434" xsi:nil="true"/>
 </relatedObjects>
 <relatingPropertyDefinition>
      <PropertySet href="i447" xsi:nil="true"/>
 </relatingPropertyDefinition>
  </RelDefinesByProperties>
```

Facility Heating Load

The criteria can be found in a *PropertySet* named **HVAC** related to the Building object. If the *PropertySingleValue* for the criteria is 0, use the PACES defaults for the associated cost model.

```
<PropertySet id="i473">
      <globalId>CIfcroot_x28514x_3212906881209983532</globalId>
      <ownerHistory>
           <OwnerHistory href="i4" xsi:nil="true"/>
      </ownerHistory>
      <name>HVAC</name>
           <PropertySingleValue>
                <name>Heating Load</name>
                <nominalValue>
                     <Text>100</Text>
                </nominalValue>
           </PropertySingleValue>
      </hasProperties>
```

```
</PropertySet>
<RelDefinesByProperties id="i474">
    <globalId>RelDefinesByProperties_xi474x_3212906881209983532</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Building href="i434" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i473" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Facility Cooling Load

The criteria can be found in a *PropertySet* named **HVAC** related to the Building object. If the *PropertySingleValue* for the criteria is 0, use the PACES defaults for the associated cost model.

```
<PropertySet id="i473">
    <globalId>CIfcroot_x28514x_3212906881209983532</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>HVAC</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Cooling Load</name>
            <nominalValue>
                <Text>10</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i474">
    <globalId>RelDefinesByProperties_xi474x_3212906881209983532</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Building href="i434" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i473" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Facility Descriptive Parameters

Soil Type

The criteria can be found in a *PropertySet* named **Earthwork** related to the Site object. If the *PropertySingleValue* for the criteria is [Not Set], use the PACES defaults for the associated cost model.

```
<PropertySet id="i400">
    <globalId>CIfcroot_x28480x_3193077074019839404</globalId>
    <ownerHistory>
```

```
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>Earthwork</name>
        <hasProperties>
            <PropertySingleValue>
                <name>Soil Type</name>
                <nominalValue>
                    <Text>[Not Set]</Text>
                </nominalValue>
            </PropertySingleValue>
        </hasProperties>
    </PropertySet>
    <RelDefinesByProperties id="i401">
        <globalId>RelDefinesByProperties_xi401x_3193077074019839404</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <relatedObjects>
            <Site href="i387" xsi:nil="true"/>
        </relatedObjects>
        <relatingPropertyDefinition>
            <PropertySet href="i400" xsi:nil="true"/>
        </relatingPropertyDefinition>
    </RelDefinesByProperties>
```

Floor Structure Type

The criteria can be found in a *PropertySet* named **Construction Type** related to the Building object. If the *PropertySingleValue* for the criteria is [Not Set], use the PACES defaults for the associated cost model.

```
    <PropertySet id="i475">
        <globalId>CIfcroot_x28515x_357314837706594610l</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>Construction Type</name>
        <hasProperties>
                <name>Floor Structure</name>
                <nominalValue>
                    <Text>Steel Fr/Lt Ld/Mtl Joist/Stl Deck/Conc Fill</Text>
                </nominalValue>
            </PropertySingleValue>
        </hasProperties>
    </PropertySet>
    <RelDefinesByProperties id="i476">
        <globalId>RelDefinesByProperties_xi476x_357314837706594610l</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <relatedObjects>
            <Building href="i434" xsi:nil="true"/>
        </relatedObjects>
        <relatingPropertyDefinition>
            <PropertySet href="i475" xsi:nil="true"/>
        </relatingPropertyDefinition>
    </RelDefinesByProperties>
```

Roof Structure Type

The criteria can be found in a *PropertySet* named **Construction Type** related to the Building object. If the *PropertySingleValue* for the criteria is [Not Set], use the PACES defaults for the associated cost model.

```
<PropertySet id="i475">
    <globalId>CIfcroot_x28515x_3573148377065946101</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Construction Type</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Roof Structure</name>
            <nominalValue>
                <Text>Steel Fr/Lt Ld/Metal Joist/Steel Roof Deck</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i476">
    <globalId>RelDefinesByProperties_xi476x_3573148377065946101</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Building href="i434" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i475" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Stair Type

The criteria can be found in a *PropertySet* named **Construction Type** related to the Building object. If the *PropertySingleValue* for the criteria is [Not Set], use the PACES defaults for the associated cost model.

```
<PropertySet id="i475">
    <globalId>CIfcroot_x28515x_3573148377065946101</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Construction Type</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Stair Type</name>
            <nominalValue>
                <Text>Concrete</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i476">
    <globalId>RelDefinesByProperties_xi476x_3573148377065946101</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
```

```
            <Building href="i434" xsi:nil="true"/>
        </relatedObjects>
        <relatingPropertyDefinition>
            <PropertySet href="i475" xsi:nil="true"/>
        </relatingPropertyDefinition>
    </RelDefinesByProperties>
```

Span Length

The criteria can be found in a *PropertySet* named **Bay** related to the Building object. If the *PropertySingleValue* for the criteria is [Not Set], simply use the PACES defaults for the associated cost model.

```
<PropertySet id="i475">
    <globalId>CIfcroot_x28515x_357314837706594610l</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Construction Type</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Span Length</name>
            <nominalValue>
                <Text>Average 30-50 LF</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i476">
    <globalId>RelDefinesByProperties_xi476x_357314837706594610l</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Building href="i434" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i475" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Foundation Type

The criteria can be found in a *PropertySet* named **Construction Type** related to the Building object. If the *PropertySingleValue* for the criteria is [Not Set], use the PACES defaults for the associated cost model.

```
<PropertySet id="i457">
    <globalId>CIfcroot_x28506x_3212906881209983532</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Foundation</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Foundation Type</name>
            <nominalValue>
                <Text>Mat Foundation</Text>
            </nominalValue>
        </PropertySingleValue>
```

```
        </hasProperties>
    </PropertySet>
    <RelDefinesByProperties id="i458">
        <globalId>RelDefinesByProperties_xi458x_3212906881209983532</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <relatedObjects>
            <Building href="i434" xsi:nil="true"/>
        </relatedObjects>
        <relatingPropertyDefinition>
            <PropertySet href="i457" xsi:nil="true"/>
        </relatingPropertyDefinition>
    </RelDefinesByProperties>
```

Roofing Type

The criteria can be found in a *PropertySet* named **Roof** related to the Building object. If the *PropertySingleValue* for the criteria is [Not Set], use the PACES defaults for the associated cost model.

```
<PropertySet id="i479">
    <globalId>CIfcroot_x28517x_357314837706594610l</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Roof</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Roofing Type</name>
            <nominalValue>
                <Text>Standing Seam Metal</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i480">
    <globalId>RelDefinesByProperties_xi480x_3573148377065946101</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Building href="i434" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i479" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>
```

Exterior Wall Type/Exterior Wall Backup

The criteria can be found in a *PropertySet* named **Construction Type** related to the Building object. If the *PropertySingleValue* for the criteria is [Not Set], use the PACES defaults for the associated cost model.

```
    <PropertySet id="i481">
        <globalId>CIfcroot_x28518x_3573148377065946101</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
```

```
            <name>Exterior Wall</name>
            <hasProperties>
                <PropertySingleValue>
                    <name>Wall Backup</name>
                    <nominalValue>
                        <Text>8 in Masonry/Mtl Furring Strip/Pnt. Gyp. Bd.</Text>
                    </nominalValue>
                </PropertySingleValue>
                <PropertySingleValue>
                    <name>Wall Type</name>
                    <nominalValue>
                        <Text>Brick Veneer</Text>
                    </nominalValue>
                </PropertySingleValue>
            </hasProperties>
        </PropertySet>
        <RelDefinesByProperties id="i482">
            <glob-
alId>RelDefinesByProperties_xi482x_357314837706594610l</globalId>
            <ownerHistory>
                <OwnerHistory href="i4" xsi:nil="true"/>
            </ownerHistory>
            <relatedObjects>
                <Building href="i434" xsi:nil="true"/>
            </relatedObjects>
            <relatingPropertyDefinition>
                <PropertySet href="i481" xsi:nil="true"/>
            </relatingPropertyDefinition>
        </RelDefinesByProperties>
```

Heat Generating System/Cooling System

The criteria can be found in a *PropertySet* named **HVAC** related to the Building object. If the *PropertySingleValue* for the criteria is [Not Set], use the PACES defaults for the associated cost model.

```
<PropertySet id="i473">
    <globalId>CIfcroot_x28514x_3212906881209983532</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>HVAC</name>
    <hasProperties>
        <PropertySingleValue>
            <name>Cooling System</name>
            <nominalValue>
                <Text>Reciprocating - Chiller</Text>
            </nominalValue>
        </PropertySingleValue>
        <PropertySingleValue>
            <name>Heat Generating System</name>
            <nominalValue>
                <Text>Hot Water Boilers - Gas Fired</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i474">
    <globalId>RelDefinesByProperties_xi474x_3212906881209983532</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <Building href="i434" xsi:nil="true"/>
```

```
</relatedObjects>
<relatingPropertyDefinition>
    <PropertySet href="i473" xsi:nil="true"/>
</relatingPropertyDefinition>
</RelDefinesByProperties>
```

Functional Space Area Parameters

FSA Code and FSA Area

The most difficult part of creating the translation process was to determine the Functional Space Area Parameters. The major issue was to first determine where to look for the criteria. This determination was accomplished by first determining if actual Space objects exist in the ifcXML. Once this existence is determined, the additional requirements are a little more self explanatory.

Spaces or no spaces

This issue was addressed by modifying the Transform ifcXML to PACES.xsl file provided by Earth Tech following the completion of the Third Contract.

ERDC-CERL added some code to EarthTech's first translation deliverable to test whether or not the FC ifcXML file contained actual spaces. If spaces do not exist, the code is to use SpaceProgram-specific information. In addition, the TypeObject related to the SpaceProgram will need to be referenced for information.

The source code below is part of the modifications to the original XSL provided by Earth Tech to determine which Space objects exist in the ifcXML file.

```
<!-- Determine which space-retrieval template to use based on the existence of space/storey relationship
-->
    <xsl:when test="$counter > 0">
    <xsl:for-each select="/ix:ifcXML/ix:RelAggregates/ix:relatingObject/ix:BuildingStorey[@href =
$storeyid]/../../ix:relatedObjects/ix:Space">
        <xsl:variable name="spaceid" select="@href"/>
        <xsl:element name="space">
            <xsl:for-each select="/ix:ifcXML/ix:RelAssignsToControl/ix:relatedObjects/ix:Space[@href
= $spaceid]/../../ix:relatingControl/ix:SpaceProgram">
                <xsl:variable name="spaceprgid" select="@href"/>
                <xsl:call-template name="EXTpropsSPACE">
                    <xsl:with-param name="spaceid" select="$spaceid"/>
                    <xsl:with-param name="spaceprgid" select="$spaceprgid"/>
                </xsl:call-template>
            </xsl:for-each>
        </xsl:element>
    </xsl:for-each>
    </xsl:when>
<xsl:otherwise>
```

```
<!-- No spaces exist so use the space program attached to the building level -->
    <xsl:for-each select="/ix:ifcXML/ix:RelAssignsToControl/ix:relatedObjects/ix:Building[@href =
$buildid]/../../ix:relatingControl/ix:SpaceProgram">
        <xsl:variable name="spaceprogid" select="@href"/>
        <xsl:for-each se-
lect="/ix:ifcXML/ix:RelDefinesByType/ix:relatedObjects/ix:SpaceProgram[@href = $space-
progid]/../../ix:relatingType/ix:TypeObject">
            <xsl:variable name="typeobjectid" select="@href"/>
            <xsl:variable name="counterpsets" select="count(/ix:ifcXML/ix:TypeObject[@id
=$typeobjectid]/ix:hasPropertySets/ix:PropertySet/ix:name[. = 'Function
Model']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']) "/>
            <xsl:if test="$counterpsets > 0">
            <xsl:element name="space">
                <xsl:call-template name="EXTpropsSPACEPROG">
                    <xsl:with-param name="typeobjid" select="$typeobjectid"/>
                    <xsl:with-param name="spaceprgid" select="$spaceprogid"/>
                </xsl:call-template>
            </xsl:element>
            </xsl:if>
        </xsl:for-each>
    </xsl:for-each>
</xsl:otherwise>

<!-- RELATED SPACE/PROGRAM PROPS -->
    <xsl:template name="EXTpropsSPACEPROG">
        <xsl:param name="typeobjid"/>
        <xsl:param name="spaceprgid"/>
        <xsl:for-each select="/ix:ifcXML/ix:TypeObject[@id
=$typeobjid]/ix:hasPropertySets/ix:PropertySet/ix:name[. = 'Function
Model']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']">
            <xsl:if test="/ix:ifcXML/ix:TypeObject[@id
=$typeobjid]/ix:hasPropertySets/ix:PropertySet/ix:name[. = 'Function
Model']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text !=
"">
                <xsl:element name="code">
                    <xsl:attribute name="fkey"><xsl:value-of select="$typeobjid"/></xsl:attribute>
                    <xsl:value-of select="/ix:ifcXML/ix:TypeObject[@id
=$typeobjid]/ix:hasPropertySets/ix:PropertySet/ix:name[. = 'Function
Model']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text"/>
                </xsl:element>
                <xsl:call-template name="EXTpropsSPACEPROGArea">
                    <xsl:with-param name="spaceid" select="$typeobjid"/>
                    <xsl:with-param name="spaceprgid" select="$spaceprgid"/>
                    <xsl:with-param name="area" select="/ix:ifcXML/ix:SpaceProgram[@id =
$spaceprgid]/ix:standardRequiredArea"/>
                </xsl:call-template>
            </xsl:if>
        </xsl:for-each>
    </xsl:template>
    <xsl:template name="EXTpropsSPACEPROGArea">
        <xsl:param name="spaceid"/>
        <xsl:param name="spaceprgid"/>
        <xsl:param name="area"/>
            <xsl:element name="area">
                <xsl:attribute name="fkey"><xsl:value-of select="$spaceid"/></xsl:attribute>
                <xsl:value-of select="$area"/>
            </xsl:element>
    </xsl:template>
    <xsl:template name="EXTpropsSPACE">
        <xsl:param name="spaceid"/>
        <xsl:param name="spaceprgid"/>
        <xsl:for-each select="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatedObjects/ix:Space[@href
=$spaceid]/../../ix:relatingPropertyDefinition/ix:PropertySet">
            <xsl:variable name="psetid" select="@href"/>
                <xsl:if test="/ix:ifcXML/ix:PropertySet[@id =
$psetid]/ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text != "">
                    <xsl:element name="code">
                        <xsl:attribute name="fkey"><xsl:value-of select="$psetid"/></xsl:attribute>
```

```
                        <xsl:value-of select="/ix:ifcXML/ix:PropertySet[@id =
$psetid]/ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text"/>
                    </xsl:element>
                    <xsl:call-template name="EXTpropsSPACEArea">
                        <xsl:with-param name="spaceid" select="$spaceid"/>
                        <xsl:with-param name="spaceprgid" select="$spaceprgid"/>
                        <xsl:with-param name="area" select="/ix:ifcXML/ix:SpaceProgram[@id =
$spaceprgid]/ix:standardRequiredArea"/>
                    </xsl:call-template>
                </xsl:if>
            </xsl:for-each>
        </xsl:template>
```

Example output from FC ifcXML

No Spaces

The example below is based on no actual spaces being present in the export file. The translator file (Earth Tech's XSL) should look at the SpaceProgram information related only to the Building object. The FSA Area will be found in the SpaceProgram, and the FSA Code will be found in the relating TypeObject.

```
<!-- ===== MDSFUNCTIONTEMPLATE : Administrative Support ===== -->
    <SpaceProgram id="i171">
        <globalId>SpaceProgram_xi171x_281742868453959512</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <spaceProgramIdentifier>Administrative Support</spaceProgramIdentifier>
        <standardRequiredArea>278.0</standardRequiredArea>
    </SpaceProgram>
    <!-- ===== 'Link' Building to the SpaceProgram ===== -->
    <RelAssignsToControl id="i172">
        <globalId>RelAssignsToControl_xi172x_281742868453959512</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <description>Relationship between Building and SpaceProgram</description>
        <relatedObjects>
            <Building href="i112" xsi:nil="true"/>
        </relatedObjects>
        <relatingControl>
            <SpaceProgram href="i171" xsi:nil="true"/>
        </relatingControl>
    </RelAssignsToControl>
    <!-- ===== 'Link' SpaceProgram to the TypeObject (which contains requirements) ===== -->
    <RelDefinesByType id="i173">
        <globalId>RelDefinesByType_xi173x_281742868453959512</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <description>Relationship between SpaceProgram and TypeObject</description>
        <relatedObjects>
            <SpaceProgram href="i171" xsi:nil="true"/>
        </relatedObjects>
        <relatingType>
            <TypeObject href="i12" xsi:nil="true"/>
        </relatingType>
    </RelDefinesByType>
    <!-- ========================================================================
-->
```

```
<!-- ===== PSET_SPACEPROGRAMCOMMON : Administrative Support ===== -->
<PropertySet id="i174">
    <globalId>PropertySet_xi174x_281742868453959512</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Pset_SpaceProgramCommon</name>
    <hasProperties>
        <PropertySingleValue>
            <name>ProgramSpaceDescription</name>
            <nominalValue>
                <Text>Administrative Support</Text>
            </nominalValue>
        </PropertySingleValue>
        <PropertySingleValue>
            <name>Function</name>
            <nominalValue>
                <Text>Administrative Support</Text>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
<RelDefinesByProperties id="i175">
    <globalId>RelDefinesByProperties_xi175x_281742868453959512</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <relatedObjects>
        <SpaceProgram href="i171" xsi:nil="true"/>
    </relatedObjects>
    <relatingPropertyDefinition>
        <PropertySet href="i174" xsi:nil="true"/>
    </relatingPropertyDefinition>
</RelDefinesByProperties>

<!-- ===== MDSFUNCTIONTEMPLATE : Administrative Support ===== -->
<TypeObject id="i12">
    <globalId>TypeObject_xi12x_1774222570673042340</globalId>
    <ownerHistory>
        <OwnerHistory href="i4" xsi:nil="true"/>
    </ownerHistory>
    <name>Administrative Support</name>
    <hasPropertySets>
        <!-- ===== MDSREQUIREMENTSET : Function Model ===== -->
        <PropertySet>
            <globalId>CIfcroot_x18874x_1774222570673042340</globalId>
            <ownerHistory>
                <OwnerHistory href="i4" xsi:nil="true"/>
            </ownerHistory>
            <name>Function Model</name>
            <hasProperties>
                <PropertySingleValue>
                    <name>Description</name>
                    <nominalValue>
                        <Text>MDS ADS ADMINISTRATIVE SUPPORT</Text>
                    </nominalValue>
                </PropertySingleValue>
                <PropertySingleValue>
                    <name>Code</name>
                    <nominalValue>
                        <Text>MAD</Text>
                    </nominalValue>
                </PropertySingleValue>
            </hasProperties>
        </PropertySet>
```

Spaces exist

In this example, the actual spaces themselves are related to a BuildingStorey object. The FSA Code information is also located within the Space object under the PropertySet "Function Model." The area to be used is also located within the Space object itself in the ElementQuantity parameter. Below is an example of such a case.

```
<!-- ===== CIFCSPACE ===== -->
    <Space id="i1704">
        <globalId>CIfcspace_x24183x_-2637240503976619207</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>Kitchen</name>
        <representation>
            <ProductDefinitionShape href="i1709" xsi:nil="true"/>
        </representation>
        <compositionType>partial</compositionType>
        <interiorOrExteriorSpace>notdefined</interiorOrExteriorSpace>
    </Space>
    <!-- ========================================================================== -->
    <!-- ===== BoundingBox ===== -->
    <BoundingBox id="i1705">
        <corner>
            <CartesianPoint>
                <coordinates>-800.0 54.0 0.0</coordinates>
            </CartesianPoint>
        </corner>
        <xDim>-20.0</xDim>
        <yDim>0.0</yDim>
        <zDim>0.0</zDim>
    </BoundingBox>
    <ShapeRepresentation id="i1706">
        <contextOfItems>
            <GeometricRepresentationContext href="i5" xsi:nil="true"/>
        </contextOfItems>
        <representationType>BoundingBox</representationType>
        <items>
            <BoundingBox href="i1705" xsi:nil="true"/>
        </items>
    </ShapeRepresentation>
    <!-- ========================================================================== -->
    <!-- ===== Space Perimeter ===== -->
    <Polyline id="i1707">
        <points>
            <CartesianPoint>
                <coordinates>-820.0 54.0 0.0</coordinates>
            </CartesianPoint>
            <CartesianPoint>
                <coordinates>-820.0 10.0 0.0</coordinates>
            </CartesianPoint>
            <CartesianPoint>
                <coordinates>-800.0 10.0 0.0</coordinates>
            </CartesianPoint>
            <CartesianPoint>
                <coordinates>-800.0 54.0 0.0</coordinates>
            </CartesianPoint>
        </points>
    </Polyline>
    <ShapeRepresentation id="i1708">
        <contextOfItems>
            <GeometricRepresentationContext href="i5" xsi:nil="true"/>
```

```
        </contextOfItems>
        <representationIdentifier>FootPrint</representationIdentifier>
        <representationType>Curve2D</representationType>
        <items>
            <Polyline href="i1707" xsi:nil="true"/>
        </items>
    </ShapeRepresentation>
    <ProductDefinitionShape id="i1709">
        <representations>
            <ShapeRepresentation href="i1706" xsi:nil="true"/>
            <ShapeRepresentation href="i1708" xsi:nil="true"/>
        </representations>
    </ProductDefinitionShape>
    <ElementQuantity id="i1710">
        <globalId>ElementQuantity_xi1710x_6847880565340129098</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <methodOfMeasurement>unknown</methodOfMeasurement>
        <quantities>
            <QuantityArea>
                <name>Area</name>
                <areaValue>880.0</areaValue>
            </QuantityArea>
        </quantities>
    </ElementQuantity>
    <RelDefinesByProperties id="i1711">
        <globalId>RelDefinesByProperties_xi1711x_6847880565340129098</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <relatedObjects>
            <Space href="i1704" xsi:nil="true"/>
        </relatedObjects>
        <relatingPropertyDefinition>
            <ElementQuantity href="i1710" xsi:nil="true"/>
        </relatingPropertyDefinition>
    </RelDefinesByProperties>

    <!-- ===== MDSREQUIREMENTSET : Function Model ===== -->
    <PropertySet id="i1712">
        <globalId>CIfcroot_x22345x_67619513994178487776</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>Function Model</name>
        <hasProperties>
            <PropertySingleValue>
                <name>Description</name>
                <nominalValue>
                    <Text>MDS KTH KITCHEN</Text>
                </nominalValue>
            </PropertySingleValue>
            <PropertySingleValue>
                <name>Code</name>
                <nominalValue>
                    <Text>MKT</Text>
                </nominalValue>
            </PropertySingleValue>
        </hasProperties>
    </PropertySet>
    <RelDefinesByProperties id="i1713">
        <globalId>RelDefinesByProperties_xi1713x_67619513994178487776</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <relatedObjects>
            <Space href="i1704" xsi:nil="true"/>
```

```
        </relatedObjects>
        <relatingPropertyDefinition>
            <PropertySet href="i1712" xsi:nil="true"/>
        </relatingPropertyDefinition>
    </RelDefinesByProperties>
```

All other space area parameters

For the values for Interior Doors, Interior Windows, Interior Overhead and Special Doors, Interior Partitions, and Interior Plumbing Fixtures are to be determined by existing PACES algorithms based on the FSA Code and FSA Areas in the xml file.

4 Mapping of Supporting Facility Requirements

Underground electrical distribution

The criteria for underground electrical distribution can be found in a *PropertySet* related to the Site object. Two *PropertySingleValues* under the *PropertySet* "Underground Electrical Distribution" are named "Length of Run" and "Primary Voltage." Length of Run will have a numerical value. If the value is 0, do not import this particular Supporting Facility Model. The user would have to either enter it manually after the importing process or use the Comparative Supporting Facilities approach as described in Chapter 5.

"Primary Voltage" will have a text value of one of the following:

5KV, 3-Phase, Primary - Concrete Encased Conduit
5KV, 3-Phase, Primary - Direct Bury
15KV, 3-Phase, Primary - Concrete Encased Conduit
15KV, 3-Phase, Primary - Direct Bury
35KV, 3-Phase, Primary - Concrete Encased Conduit
70KV, 3-Phase, Primary - Concrete Encased Conduit
Underground Residential
[Not Set]

If [Not Set] is selected, simply ignore this supporting facility model.

```
<!-- ===== MDSREQUIREMENTSET : Underground Electrical Distribution ===== -->
    <PropertySet id="i64">
        <globalId>CIfcroot_x462x_3681004551931801453</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>Underground Electrical Distribution</name>
        <hasProperties>
            <PropertySingleValue>
                <name>Length of Run</name>
                <nominalValue>
                    <Real>0.0</Real>
                </nominalValue>
            </PropertySingleValue>
            <PropertySingleValue>
                <name>Primary Voltage</name>
                <nominalValue>
                    <Text>[Not Set]</Text>
                </nominalValue>
            </PropertySingleValue>
```

```
        </hasProperties>
    </PropertySet>
    <RelDefinesByProperties id="i65">
        <globalId>RelDefinesByProperties_xi65x_3681004551931801453</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <relatedObjects>
            <Site href="i33" xsi:nil="true"/>
        </relatedObjects>
        <relatingPropertyDefinition>
            <PropertySet href="i64" xsi:nil="true"/>
        </relatingPropertyDefinition>
    </RelDefinesByProperties>
```

Parking lots

To import Parking Lot information, the following criteria must be in the ifcXML file. Two criteria values must be selected for the importing of Parking Lots to be valid. “Type of Surface” should be selected as either Asphalt or Gravel. If the value is set to [Not Set], this parameter has not been fulfilled. Also, the value for either “Number of Spaces” or “Pavement Area” must be greater than zero.

If the above criteria are not valid, simply do not import this particular Supporting Facility Model. The user would either enter it manually after the importing process or use the Comparative Supporting Facilities approach.

```
<!-- ===== MDSREQUIREMENTSET : Parking Lots ===== -->
    <PropertySet id="i84">
        <globalId>CIfcroot_x472x_3671089648336729389</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>Parking Lots</name>
        <hasProperties>
            <PropertySingleValue>
                <name>Number of spaces</name>
                <nominalValue>
                    <Label>0.0</Label>
                </nominalValue>
            </PropertySingleValue>
            <PropertySingleValue>
                <name>Pavement Area</name>
                <nominalValue>
                    <AreaMeasure>0.0</AreaMeasure>
                </nominalValue>
            </PropertySingleValue>
            <PropertySingleValue>
                <name>Type of Surface</name>
                <nominalValue>
                    <Text>Asphalt</Text>
                </nominalValue>
            </PropertySingleValue>
        </hasProperties>
    </PropertySet>
    <RelDefinesByProperties id="i85">
        <globalId>RelDefinesByProperties_xi85x_3671089648336729389</globalId>
```

```
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <relatedObjects>
            <Site href="i33" xsi:nil="true"/>
        </relatedObjects>
        <relatingPropertyDefinition>
            <PropertySet href="i84" xsi:nil="true"/>
        </relatingPropertyDefinition>
    </RelDefinesByProperties>
```

Fencing

To import Fencing information, the following criteria must be in the ifcXML file related to the IfcSite object. Both the Fencing Type and Fencing Length must be identified for the import of this Supporting Facility type to be valid. “Type of Fence” should be selected as either Boundary, Security, or Privacy. If the value is set to [Not Set], this parameter has not been fulfilled. Also, the value for “Fencing Length” must be a value greater than 0 or it is not valid.

If the above criteria are not valid, simply do not import this particular Supporting Facility Model. The user would either enter it manually after the importing process or use the Comparative Supporting Facilities approach.

```
<!-- ===== MDSREQUIREMENTSET : Physical Security ===== -->
    <PropertySet id="i586">
        <globalId>CIfcroot_x33316x_7382553245963880301</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>Physical Security</name>
        <hasProperties>
            <PropertySingleValue>
                <name>Fence Length</name>
                <nominalValue>
                    <Real>200</Real>
                </nominalValue>
                <unit>
                    <ConversionBasedUnit>
                        <dimensions>
                            <DimensionalExponents>
                                <lengthExponent>1</lengthExponent>
                                <massExponent>0</massExponent>
                                <timeExponent>0</timeExponent>
                                <electricCurrentExponent>0</electricCurrentExponent>
                                <thermodynamicTemperatureExpo-
nent>0</thermodynamicTemperatureExponent>
                                <amountOfSubstanceExponent>0</amountOfSubstanceExponent>
                                <luminousIntensityExponent>0</luminousIntensityExponent>
                            </DimensionalExponents>
                        </dimensions>
                        <unitType>lengthunit</unitType>
                        <name>lf</name>
                        <conversionFactor>
                            <MeasureWithUnit>
                                <valueComponent>
```

```
                                    <Real>304.8</Real>
                                </valueComponent>
                                <unitComponent>
                                    <SIUnit href="i7" xsi:nil="true"/>
                                </unitComponent>
                            </MeasureWithUnit>
                        </conversionFactor>
                    </ConversionBasedUnit>
                </unit>
            </PropertySingleValue>
            <PropertySingleValue>
                <name>Fence Type</name>
                <nominalValue>
                    <Text>Boundry</Text>
                </nominalValue>
            </PropertySingleValue>
        </hasProperties>
    </PropertySet>
```

Water distribution

To import Water Distribution information, the following criteria must be in the ifcXML file related to the IfcSite object. Both the Pipe Size and Length of Run must be identified for the import of this Supporting Facility type to be valid. Also, the value for both “Pipe Size” and “Length of Run” must be greater than 0; otherwise, it is not valid. “Load and Haul” value will always be present as “false.”

If the above criteria are not valid, do not import this particular Supporting Facility Model. The user would either enter it manually after the importing process or use the Comparative Supporting Facilities approach.

```
<!-- ===== MDSREQUIREMENTSET : Water Distribution ===== -->
    <PropertySet id="i570">
        <globalId>CIfcroot_x33308x_279691030103079586 8</globalId>
        <ownerHistory>
            <OwnerHistory href="i4" xsi:nil="true"/>
        </ownerHistory>
        <name>Water Distribution</name>
        <hasProperties>
            <PropertySingleValue>
                <name>Length of Run</name>
                <nominalValue>
                    <Real>50</Real>
                </nominalValue>
                <unit>
                    <ConversionBasedUnit>
                        <dimensions>
                            <DimensionalExponents>
                                <lengthExponent>1</lengthExponent>
                                <massExponent>0</massExponent>
                                <timeExponent>0</timeExponent>
                                <electricCurrentExponent>0</electricCurrentExponent>
                                <thermodynamicTemperatureExpo-
nent>0</thermodynamicTemperatureExponent>
                                <amountOfSubstanceExponent>0</amountOfSubstanceExponent>
                                <luminousIntensityExponent>0</luminousIntensityExponent>
                            </DimensionalExponents>
                        </dimensions>
```

```
                    <unitType>lengthunit</unitType>
                    <name>lf</name>
                    <conversionFactor>
                        <MeasureWithUnit>
                            <valueComponent>
                                <Real>304.8</Real>
                            </valueComponent>
                            <unitComponent>
                                <SIUnit href="i7" xsi:nil="true"/>
                            </unitComponent>
                        </MeasureWithUnit>
                    </conversionFactor>
                </ConversionBasedUnit>
            </unit>
        </PropertySingleValue>
        <PropertySingleValue>
            <name>Pipe Size</name>
            <nominalValue>
                <LengthMeasure>6</LengthMeasure>
            </nominalValue>
        </PropertySingleValue>
        <PropertySingleValue>
            <name>Load and Haul</name>
            <nominalValue>
                <Boolean>false</Boolean>
            </nominalValue>
        </PropertySingleValue>
    </hasProperties>
</PropertySet>
```

5 Mapping of Comparative Supporting Facilities

Comparative Supporting Facilities

The criteria can be found in a *PropertySet* named **Comparative Supporting Facilities** related to the Building object. Five different *PropertySingleValues* are associated with this *PropertySet*. They are Use, Facility Percentage, Pavement, Site Improvements, and Utilities.

For Pavement, Site Improvements, and Utilities, not all three need to be entered for the import to be valid. If only one has a valid value, simply average the final two so that the total of all three is 100 percent.

Use

The first item that needs to be determined is if Comparative Supporting Facilities capability should be used. This criterion is simply a Boolean condition.

If the *PropertySingleValue* **Use** is "true," start by using the PACES defaults and continue to search to see if any other values related to Comparative Supporting Facilities are defined. If false, ignore during import.

```
<PropertySingleValue>
    <name>Use</name>
    <nominalValue>
        <Boolean>true</Boolean>
    </nominalValue>
</PropertySingleValue>
```

Facility Percentage

This variable should be looked at only if the Comparative Supporting Facility Use is "true."

If the *PropertySingleValue* **Facility Percentage** is [Not Set], ignore this and use defaults.

```
<PropertySingleValue>
    <name>Facility Percentage</name>
    <nominalValue>
        <Text>[Not Set]</Text>
```

```
    </nominalValue>
</PropertySingleValue>
```

Pavement

This variable should be looked at only if the Comparative Supporting Facility Use is “true.”

If the *PropertySingleValue* **Pavement** is [Not Set], ignore it and use defaults.

```
<PropertySingleValue>
    <name>Pavements</name>
    <nominalValue>
        <Text>[Not Set]</Text>
    </nominalValue>
</PropertySingleValue>
```

Site improvements

This should be looked at only if the Comparative Supporting Facility Use is “true.”

If the *PropertySingleValue* **Site Improvements** is [Not Set], ignore it and use defaults.

```
<PropertySingleValue>
    <name>Site Improvements</name>
    <nominalValue>
        <Text>[Not Set]</Text>
    </nominalValue>
</PropertySingleValue>
```

Utilities

This variable should be looked at only if the Comparative Supporting Facility Use is “true.”

If the *PropertySingleValue* **Utilities** is [Not Set], ignore it and use defaults.

```
<PropertySingleValue>
    <name>Utilities</name>
    <nominalValue>
        <Text>[Not Set]</Text>
    </nominalValue>
</PropertySingleValue>
```

Appendix A: Final Translation Files

Final XML Translation File (XSL)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by Sean Martin (Earth Tech) -->
<!-- PACES™ 2005 XML Import Translator v1.5 - 060125 - ©2006  Earth Tech, Inc. For use with CERL (Building Composer) ifcXML data files-->
<!-- For information on CERL's Building Composer suite please see:  https://ff.cecer.army.mil/fc/ -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:ix="http://www.iai-international.org/ifcXML/ifc2x/IFC2X_FINAL">
	<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
<!-- ifxXML Data is broken down as follows:		-->
<!--	Project				-->
<!--		Site			-->
<!--		Building		-->
<!--			Building Storey		-->
<!--				Space		-->
<!--					Space Program		-->
<!-- PACES Data is broken down as follows:		-->
<!--	Project				-->
<!--		Facility			-->
<!--		Functional Space Area		-->
<!-- This XSL file transforms a ifcXML file to a	-->
<!-- PACES format by merging the ifcXML Project, -->
<!-- and Site into the PACES Project.		-->
<!-- Merging the Building and Building Storey data -->
<!-- into the PACES Facility.			-->
<!-- Merging the Space and Space program props -->
<!-- into PACES FSA data.			-->
<!-- It is important to note that CERL include PACES -->
<!-- properties in their export files and that these-->
<!-- properties are not a standard set of the ifcXML-->
<!-- specification. However this extensibility is	-->
<!-- allowed in the spec.			-->
  <xsl:template match="/">
    <!-- <earthtech_paces> -->
    <xsl:element name="earthtech_paces">
      <xsl:for-each select="/ix:ifcXML/ix:Project">
        <!-- <project> -->
        <xsl:element name="project">
          <xsl:variable name="projid" select="@id"/>
          <xsl:variable name="owneridhist" select=."/ix:ownerHistory/ix:OwnerHistory/@href"/>
          <xsl:variable name="siteids" select="/ix:ifcXML/ix:RelAggregates/ix:relatingObject/ix:Project[@href = $projid]/../../ix:relatedObjects/ix:Site/@href"/>
          <xsl:variable name="buildids" select="/ix:ifcXML/ix:RelAggregates/ix:relatingObject/ix:Site[@href = $siteids]/../../ix:relatedObjects/ix:Building/@href"/>
          <!-- *Project Properties -->
          <xsl:call-template name="propsPROJECT">
            <xsl:with-param name="projid" select="$projid"/>
            <xsl:with-param name="owneridhist" select="$owneridhist"/>
            <xsl:with-param name="siteid" select="/ix:ifcXML/ix:Site/@id"/>
          </xsl:call-template>
          <xsl:element name="markups"/>
          <!--* Sitework Models-->
          <xsl:element name="siteworks">
            <xsl:call-template name="sitework">
              <xsl:with-param name="projid" select="$projid"/>
              <xsl:with-param name="siteids" select="$siteids"/>
            </xsl:call-template>
          </xsl:element>
          <!-- <facilities> -->
          <xsl:element name="facilities">
            <xsl:for-each select="/ix:ifcXML/ix:Building[@id = $buildids]">
              <!-- <facility> -->
              <xsl:element name="facility">
```

```
                        <xsl:variable name="buildid" select="@id"/>
                        <xsl:variable name="storeyids" select="/ix:ifcXML/ix:RelAggregates/ix:relatingObject/ix:Building[@href =
$buildid]/../../ix:relatedObjects/ix:BuildingStorey/@href"/>
                        <xsl:call-template name="propsBUILDING">
                           <xsl:with-param name="buildid" select="$buildid"/>
                           <xsl:with-param name="projid" select="$projid"/>
                           <xsl:with-param name="siteids" select="$siteids"/>
                        </xsl:call-template>
                        <xsl:call-template name="calcstrelv">
                           <xsl:with-param name="buildid" select="$buildid"/>
                           <xsl:with-param name="storeyids" select="$storeyids"/>
                        </xsl:call-template>
                        <xsl:call-template name="calcfloor">
                           <xsl:with-param name="buildid" select="$buildid"/>
                        </xsl:call-template>
                        <!-- <spaces> -->
                        <xsl:element name="spaces">
                           <!--xsl:if test ="1=2"-->
                           <xsl:variable name="spaceids" select="/ix:ifcXML/ix:RelAggregates/ix:relatingObject/ix:BuildingStorey[@href
= $storeyids]/../../ix:relatedObjects/ix:Space/@href"/>
                           <xsl:choose>
                              <!-- *If the Spaces have been placed in BC(Layout), then they will be attached to the Building Storey -->
                              <xsl:when test="count($spaceids)">
                                 <xsl:variable name="spaceprgids" se-
lect="/ix:ifcXML/ix:RelAssignsToControl/ix:relatedObjects/ix:Space[@href =
$spaceids]/../../ix:relatingControl/ix:SpaceProgram/@href"/>
                                 <xsl:for-each select="$spaceprgids">
                                    <!-- <space> -->
                                    <xsl:element name="space">
                                       <xsl:variable name="spaceid" select=."/../../../ix:relatedObjects/ix:Space/@href"/>
                                       <xsl:variable name="psetid" se-
lect="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatedObjects/ix:Space[@href
=$spaceid]/../../ix:relatingPropertyDefinition/ix:PropertySet/@href"/>
                                       <xsl:if test="/ix:ifcXML/ix:PropertySet[@id =
$psetid]/ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text != "">
                                          <xsl:element name="code">
                                             <xsl:attribute name="fkey"><xsl:value-of select="$psetid"/></xsl:attribute>
                                             <xsl:value-of select="/ix:ifcXML/ix:PropertySet[@id =
$psetid]/ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text"/>
                                          </xsl:element>
                                          <xsl:call-template name="EXTpropsSPACEArea">
                                             <xsl:with-param name="spaceid" select="$spaceid"/>
                                             <xsl:with-param name="spaceprgid" select=.""/>
                                             <xsl:with-param name="area" select="/ix:ifcXML/ix:SpaceProgram[@id =
.]/ix:standardRequiredArea"/>
                                          </xsl:call-template>
                                       </xsl:if>
                                    </xsl:element>
                                 </xsl:for-each>
                              </xsl:when>
                              <!-- *If the Spaces have NOT been placed in BC, then they will be attached to the Building Itself. They are
only defined as requirements -->
                              <xsl:otherwise>
                                 <xsl:variable name="spaceprgids" se-
lect="/ix:ifcXML/ix:RelAssignsToControl/ix:relatedObjects/ix:Building[@href =
$buildid]/../../ix:relatingControl/ix:SpaceProgram/@href"/>
                                 <xsl:variable name="typeobjectids" se-
lect="/ix:ifcXML/ix:RelDefinesByType/ix:relatedObjects/ix:SpaceProgram[@href =
$spaceprgids]/../../ix:relatingType/ix:TypeObject/@href"/>
                                 <!-- *Hunt for Cost Model Data Using 'Functional Model' PSET then verify Function Model has a Code --
>
                                 <xsl:for-each select="$typeobjectids">
                                    <xsl:variable name="typeobjectid" select=.""/>
                                    <xsl:variable name="spaceprgid" select=."/../../../ix:relatedObjects/ix:SpaceProgram/@href"/>
                                    <xsl:if test="/ix:ifcXML/ix:TypeObject[@id =$typeobjectid]/ix:hasPropertySets/ix:PropertySet/ix:name[.
= 'Function Model']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text != "">
                                       <!-- <space> -->
                                       <xsl:element name="space">
                                          <xsl:element name="code">
                                             <xsl:attribute name="fkey"><xsl:value-of select=""/></xsl:attribute>
```

```
                                        <xsl:value-of select="/ix:ifcXML/ix:TypeObject[@id
=$typeobjectid]/ix:hasPropertySets/ix:PropertySet/ix:name[. = 'Function Model']/../ix:hasProperties/ix:PropertySingleValue/ix:name[.
= 'Code']/../ix:nominalValue/ix:Text"/>
                                    </xsl:element>
                                    <xsl:element name="area">
                                        <xsl:value-of select="/ix:ifcXML/ix:SpaceProgram[@id =
$spaceprgid]/ix:standardRequiredArea"/>
                                    </xsl:element>
                                </xsl:element>
                            </xsl:if>
                        </xsl:for-each>
                    </xsl:otherwise>
                </xsl:choose>
                <!--/xsl:if-->
            </xsl:element>
        </xsl:element>
    </xsl:for-each>
</xsl:element>
</xsl:element>
</xsl:for-each>
</xsl:element>
</xsl:template>
<!-- ============= -->
<!-- ============= -->
<!-- ================= -->
<!-- ===LEVEL TEMPLATES=== -->
<!-- ================= -->
<!-- ============= -->
<!-- ============= -->
<!-- LOCAL PROJECT PROPS -->
<xsl:template name="propsPROJECT">
    <xsl:param name="projid"/>
    <xsl:param name="owneridhist"/>
    <xsl:param name="siteid"/>

    <xsl:for-each select="/ix:ifcXML/ix:Project[@id =$projid]">
        <xsl:if test="./ix:name != ''">
            <xsl:element name="name">
                <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
                <xsl:value-of select="./ix:name"/>
            </xsl:element>
        </xsl:if>

        <xsl:if test="./ix:longName != ''">
            <xsl:element name="description">
                <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
                <xsl:value-of select="./ix:longName"/>
            </xsl:element>
        </xsl:if>

        <xsl:if test="./ix:description != ''">
            <xsl:element name="comments">
                <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
                <xsl:value-of select="./ix:description"/>
            </xsl:element>
        </xsl:if>

        <xsl:if test="./ix:unitsincontext != ''">
            <xsl:element name="units">
                <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
                <xsl:value-of select="./ix:unitsincontext"/>
            </xsl:element>
        </xsl:if>
    </xsl:for-each>

    <!-- *'Pset_CCProjectData' is a location CERL has set aside for Project Date/Data specific elements  -->
    <xsl:for-each select="/ix:ifcXML/ix:PropertySet/ix:name[. = 'Pset_CCProjectData']">
        <xsl:variable name="psetid" select="./@id"/>

        <xsl:if test="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Project[@href =$projid]/@href  != ''">
            <xsl:if test="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Number']/../ix:nominalValue/ix:Integer != ''">
```

```
            <xsl:element name="number">
               <xsl:attribute name="fkey"><xsl:value-of select=".”./@id"/></xsl:attribute>
               <xsl:value-of select=".”./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Number']/../ix:nominalValue/ix:Integer"/>
            </xsl:element>
         </xsl:if>

         <xsl:if test=".”./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'PreparedBy']/../ix:nominalValue/ix:Text != "">
            <xsl:element name="prepared_by">
               <xsl:attribute name="fkey"><xsl:value-of select=".”./@id"/></xsl:attribute>
               <xsl:value-of select=".”./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Pre-
paredBy']/../ix:nominalValue/ix:Text"/>
            </xsl:element>
         </xsl:if>

         <xsl:if test=".”./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'PreparedDate']/../ix:nominalValue/ix:Text != "">
            <xsl:element name="date_prepared">
               <xsl:attribute name="fkey"><xsl:value-of select=".”./@id"/></xsl:attribute>
               <xsl:value-of select=".”./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Prepared-
Date']/../ix:nominalValue/ix:Text"/>
            </xsl:element>
         </xsl:if>

         <xsl:if test="string(number(../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Fiscal']/../ix:nominalValue/ix:Integer)) !=
'NaN'">
            <xsl:element name="program_year">
               <xsl:attribute name="fkey"><xsl:value-of select=".”./@id"/></xsl:attribute>
               <xsl:value-of select=".”./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Fiscal']/../ix:nominalValue/ix:Integer "/>
            </xsl:element>
         </xsl:if>

      </xsl:if>
   </xsl:for-each>

   <xsl:for-each select="/ix:ifcXML/ix:OwnerHistory[@id = $owneridhist]">
      <xsl:variable name="persorgid" select=".”/ix:owningUser/ix:PersonAndOrganization/@href"/>

      <xsl:if test=".”/ix:CreationDate != "">
         <xsl:element name="date_prepared">
            <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
            <xsl:value-of select=".”/ix:CreationDate "/>
         </xsl:element>
      </xsl:if>

      <xsl:for-each select="/ix:ifcXML/ix:PersonAndOrganization[@id = $persorgid]">
         <xsl:variable name="persid" select=".”/ix:thePerson/ix:Person/@href"/>
         <xsl:for-each select="/ix:ifcXML/ix:Person[@id = $persid]">
            <xsl:if test=".”/ix:FamilyName != "">
               <xsl:element name="prepared_by">
                  <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
                  <xsl:value-of select=".”/ix:FamilyName"/>
               </xsl:element>
            </xsl:if>
         </xsl:for-each>
      </xsl:for-each>
   </xsl:for-each>

   <xsl:for-each select="/ix:ifcXML/ix:Site[@id = $siteid]">
      <xsl:if test=".”/ix:siteAddress/ix:PostalAddress/ix:region  != "">
         <xsl:element name="region">
            <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
            <xsl:value-of select=".”/ix:siteAddress/ix:PostalAddress/ix:region"/>
         </xsl:element>
      </xsl:if>

      <xsl:if test=".”/ix:siteAddress/ix:PostalAddress/ix:town != "">
         <xsl:element name="location">
            <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
            <xsl:value-of select=".”/ix:siteAddress/ix:PostalAddress/ix:town"/>
         </xsl:element>
      </xsl:if>

      <xsl:if test=".”/ix:siteAddress/ix:PostalAddress/ix:postalCode!= "">
```

```
            <xsl:element name="zip_code">
               <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
               <xsl:value-of select=."/ix:siteAddress/ix:PostalAddress/ix:postalCode"/>
            </xsl:element>
         </xsl:if>

      </xsl:for-each>
   </xsl:template>
   <!-- ================= -->
   <!-- ================= -->
   <!-- ================= -->
   <!-- ================= -->
   <!-- FACILITY PROPS -->
   <xsl:template name="propsBUILDING">
      <xsl:param name="buildid"/>
      <xsl:param name="projid"/>
      <xsl:param name="siteids"/>

      <xsl:for-each select="/ix:ifcXML/ix:Building[@id = $buildid]">
         <xsl:if test=."/ix:name != ''">
            <xsl:element name="name">
               <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
               <xsl:value-of select=."/ix:name"/>
            </xsl:element>
         </xsl:if>

         <xsl:if test=."/ix:longname != ''">
            <xsl:element name="description">
               <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
               <xsl:value-of select=."/ix:longname"/>
            </xsl:element>
         </xsl:if>

         <xsl:if test=."/ix:description != ''">
            <xsl:element name="comments">
               <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
               <xsl:value-of select=."/ix:description"/>
            </xsl:element>
         </xsl:if>

      </xsl:for-each>

      <xsl:variable name="storeyids" se-
lect="/ix:ifcXML/ix:RelAggregates/ix:relatingObject/ix:Building[@href=$buildid]/../../ix:relatedObjects/ix:BuildingStorey/@href"/>
      <!-- *Get Floor2Floor and Floor2Ceiling -->
      <xsl:call-template name="StoreyHeights">
         <xsl:with-param name="buildid" select="$buildid"/>
         <xsl:with-param name="storeyidsag" select="/ix:ifcXML/ix:BuildingStorey[@id = $storeyids]/ix:elevation[. &gt;= 0]/../@id"/>
         <xsl:with-param name="storeyidsab" select="/ix:ifcXML/ix:BuildingStorey[@id = $storeyids]/ix:elevation[. &lt; 0]/../@id"/>
      </xsl:call-template>

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet/ix:name[. = 'Pset_CCBuildingData']">
         <xsl:variable name="psetid" select=."/@id"/>

         <xsl:if test="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href  != ''">
            <!-- *Use GrossActualArea Over GrossPlanedArea -->
            <xsl:choose>
               <xsl:when test="number(../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'GrossActu-
alArea']/../ix:nominalValue/ix:Real) &gt; 0">
                  <xsl:element name="gross_area">
                     <xsl:attribute name="fkey"><xsl:value-of select=."/@id"/></xsl:attribute>
                     <xsl:value-of select=."/ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'GrossActu-
alArea']/../ix:nominalValue/ix:Real"/>
                  </xsl:element>
               </xsl:when>

               <xsl:when test="number(../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'GrossPlanne-
dArea']/../ix:nominalValue/ix:Real) &gt; 0">
                  <xsl:element name="gross_area">
                     <xsl:attribute name="fkey"><xsl:value-of select=."/@id"/></xsl:attribute>
```

```
                    <xsl:value-of select=”./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'GrossPlanne-
dArea']/../ix:nominalValue/ix:Real"/>
                </xsl:element>
            </xsl:when>

            <xsl:otherwise>
                <xsl:element name="gross_area">
                    <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
                    <xsl:value-of select="0"/>
                </xsl:element>
            </xsl:otherwise>

        </xsl:choose>
    </xsl:if>
</xsl:for-each>

<!-- *The PACES property sets are included by CERL and are not a part of the ifcXML spec. -->
<xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'Earthwork']/ix:hasProperties/ix:PropertySingleValue[./ix:name =
'Soil Type']/ix:nominalValue/ix:Text/../../../../@id">
    <xsl:variable name="psetid" select=.""/>
    <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Site[@href =$siteids]/@href)">
        <xsl:element name="soil_type">
            <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
            <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Soil Type']/ix:nominalValue/ix:Text"/>
        </xsl:element>
    </xsl:if>
</xsl:for-each>

<xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'HVAC']/ix:hasProperties/ix:PropertySingleValue[./ix:name =
'Cooling System']/ix:nominalValue/ix:Text/../../../../@id">
    <xsl:variable name="psetid" select=.""/>
    <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
        <xsl:element name="cooling_sys">
            <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
            <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Cooling Sys-
tem']/ix:nominalValue/ix:Text"/>
        </xsl:element>
    </xsl:if>
</xsl:for-each>

<xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'HVAC']/ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Heat
Generating System']/ix:nominalValue/ix:Text/../../../../@id">
    <xsl:variable name="psetid" select=.""/>
    <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
        <xsl:element name="heat_gen_sys">
            <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
            <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Heat Generating Sys-
tem']/ix:nominalValue/ix:Text"/>
        </xsl:element>
    </xsl:if>
</xsl:for-each>
<xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'Loads']/ix:hasProperties/ix:PropertySingleValue[./ix:name =
'Electrical Loads']/ix:nominalValue/ix:Text/../../../../@id">
    <xsl:variable name="psetid" select=.""/>
    <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
        <xsl:element name="electrical_load">
            <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
            <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Electrical
Loads']/ix:nominalValue/ix:Text"/>
        </xsl:element>
    </xsl:if>
</xsl:for-each>

<xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'Construction
Type']/ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Floor Structure']/ix:nominalValue/ix:Text/../../../../@id">
    <xsl:variable name="psetid" select=.""/>
    <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
```

```
            <xsl:element name="floor_structure">
               <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
               <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Floor Struc-
ture']/ix:nominalValue/ix:Text"/>
            </xsl:element>
         </xsl:if>
      </xsl:for-each>

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'Construction
Type']/ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Roof Structure']/ix:nominalValue/ix:Text/../../../../@id">
         <xsl:variable name="psetid" select=.""/>
         <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
            <xsl:element name="roof_structure">
               <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
               <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Roof Structure']/ix:nominalValue/ix:Text"/>
            </xsl:element>
         </xsl:if>
      </xsl:for-each>

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'Construction
Type']/ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Stair Type']/ix:nominalValue/ix:Text/../../../../@id">
         <xsl:variable name="psetid" select=.""/>
         <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
            <xsl:element name="stair_type">
               <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
               <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Stair Type']/ix:nominalValue/ix:Text"/>
            </xsl:element>
         </xsl:if>
      </xsl:for-each>

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'Bay']/ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Span
Length']/ix:nominalValue/ix:Text/../../../../@id">
         <xsl:variable name="psetid" select=.""/>
         <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
            <xsl:element name="bay_type">
               <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
               <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Span Length']/ix:nominalValue/ix:Text"/>
            </xsl:element>
         </xsl:if>
      </xsl:for-each>

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'Construction
Type']/ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Foundation Type']/ix:nominalValue/ix:Text/../../../../@id">
         <xsl:variable name="psetid" select=.""/>
         <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
            <xsl:element name="foundation_type">
               <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
               <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Foundation
Type']/ix:nominalValue/ix:Text"/>
            </xsl:element>
         </xsl:if>
      </xsl:for-each>

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'Roof']/ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Roofing
Type']/ix:nominalValue/ix:Text/../../../../@id">
         <xsl:variable name="psetid" select=.""/>
         <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
            <xsl:element name="roofing_type">
               <xsl:attribute name="fkey"><xsl:value-of select=”./@id"/></xsl:attribute>
               <xsl:value-of select=”/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Roofing Type']/ix:nominalValue/ix:Text"/>
            </xsl:element>
         </xsl:if>
      </xsl:for-each>

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[ix:name='Construction Type' and
ix:hasProperties/ix:PropertySingleValue/ix:name = 'Wall Backup' and ix:hasProperties/ix:PropertySingleValue/ix:name = 'Wall
Type']/@id">
```

```
        <xsl:variable name="psetid" select="."/>
        <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
          <xsl:element name="walls">
            <xsl:element name="wall">
              <xsl:element name="type">
                <xsl:attribute name="fkey"><xsl:value-of select="$psetid"/></xsl:attribute>
                <xsl:value-of select="/ix:ifcXML/ix:PropertySet[@id = $psetid]/ix:hasProperties/ix:PropertySingleValue[./ix:name =
'Wall Type']/ix:nominalValue/ix:Text"/>

              </xsl:element>
              <xsl:element name="backup">
                <xsl:attribute name="fkey"><xsl:value-of select="$psetid"/></xsl:attribute>
                <xsl:value-of select="/ix:ifcXML/ix:PropertySet[@id = $psetid]/ix:hasProperties/ix:PropertySingleValue[./ix:name =
'Wall Backup']/ix:nominalValue/ix:Text"/>
              </xsl:element>
              <xsl:element name="percent">
                <xsl:attribute name="fkey"><xsl:value-of select="$psetid"/></xsl:attribute>
                <xsl:value-of select="1"/>
              </xsl:element>
            </xsl:element>
          </xsl:element>
        </xsl:if>
      </xsl:for-each>


      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'HVAC']/ix:hasProperties/ix:PropertySingleValue[./ix:name =
'Heating Load']/ix:nominalValue/ix:Text/../../../../@id">
        <xsl:variable name="psetid" select="."/>
        <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
          <xsl:element name="heating_load">
            <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
            <xsl:value-of select="/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Heating Load']/ix:nominalValue/ix:Text"/>
          </xsl:element>
        </xsl:if>
      </xsl:for-each>

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[./ix:name = 'HVAC']/ix:hasProperties/ix:PropertySingleValue[./ix:name =
'Cooling Load']/ix:nominalValue/ix:Text/../../../../@id">
        <xsl:variable name="psetid" select="."/>
        <xsl:if test="count(/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href)">
          <xsl:element name="cooling_load">
            <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
            <xsl:value-of select="/../ix:hasProperties/ix:PropertySingleValue[./ix:name = 'Cooling Load']/ix:nominalValue/ix:Text"/>
          </xsl:element>
        </xsl:if>
      </xsl:for-each>

      <!-- ************************-->
      <xsl:for-each select="/ix:ifcXML/ix:PropertySet/ix:name[. = 'PACES']">
        <xsl:variable name="psetid" select="./@id"/>
        <xsl:if test="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href  != ''">
          <xsl:if test="string(number(../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Facility Foot
Print']/../ix:nominalValue/ix:Real)) != 'NaN'">
            <xsl:element name="footprint">
              <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
              <xsl:value-of select="../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Facility Foot
Print']/../ix:nominalValue/ix:Real"/>
            </xsl:element>
          </xsl:if>

          <xsl:if test="string(number(../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Facility Perime-
ter']/../ix:nominalValue/ix:Real)) != 'NaN'">
            <xsl:element name="perimeter">
              <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
              <xsl:value-of select="../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Facility Perime-
ter']/../ix:nominalValue/ix:Real"/>
            </xsl:element>
          </xsl:if>
```

```
            <xsl:if test="string(number(../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Exterior
Doors']/../ix:nominalValue/ix:Integer)) != 'NaN'">
               <xsl:element name="ext_doors">
                  <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
                  <xsl:value-of select="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Exterior
Doors']/../ix:nominalValue/ix:Integer"/>
               </xsl:element>
            </xsl:if>
         </xsl:if>
      </xsl:for-each>

      <!-- Comparative Supporting Facilities -->
      <xsl:for-each select="/ix:ifcXML/ix:PropertySet/ix:name[. = 'Comparative Supporting Facilities']">
         <xsl:variable name="psetid" select="./@id"/>
         <xsl:if test="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href  != ''">

            <xsl:if test="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Use']/../ix:nominalValue/ix:Boolean != ''">
               <xsl:element name="use">
                  <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
                  <xsl:value-of select="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Use']/../ix:nominalValue/ix:Boolean"/>
               </xsl:element>
            </xsl:if>

            <xsl:if test="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Facility Percentage']/../ix:nominalValue/ix:Text != ''">
               <xsl:element name="facility_percentage">
                  <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
                  <xsl:value-of select="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Facility Percent-
age']/../ix:nominalValue/ix:Text"/>
               </xsl:element>
            </xsl:if>

            <xsl:if test="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Pavements']/../ix:nominalValue/ix:Text != ''">
               <xsl:element name="pavements">
                  <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
                  <xsl:value-of select="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Pavements']/../ix:nominalValue/ix:Text"/>
               </xsl:element>
            </xsl:if>

            <xsl:if test="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Site Improvements']/../ix:nominalValue/ix:Text != ''">
               <xsl:element name="site_improvements">
                  <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
                  <xsl:value-of select="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Site Improve-
ments']/../ix:nominalValue/ix:Text"/>
               </xsl:element>
            </xsl:if>

            <xsl:if test="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Utilities']/../ix:nominalValue/ix:Text != ''">
               <xsl:element name="utilities">
                  <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
                  <xsl:value-of select="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Utilities']/../ix:nominalValue/ix:Text"/>
               </xsl:element>
            </xsl:if>
         </xsl:if>
      </xsl:for-each>

      <!-- *Hunt for Cost Model Data Using 'Cost Model' PSET then verify matching specific Model to our Current Building -->
      <xsl:for-each select="/ix:ifcXML/ix:PropertySet/ix:name[. = 'Cost Models']">
         <xsl:variable name="psetid" select="./@id"/>
         <xsl:if test="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatingPropertyDefinition/ix:PropertySet[@href =
$psetid]/../../ix:relatedObjects/ix:Building[@href =$buildid]/@href  != ''">
            <xsl:if test="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Model Code']/../ix:nominalValue/ix:Text != ''">
               <xsl:element name="model_code">
                  <xsl:attribute name="fkey"><xsl:value-of select="./@id"/></xsl:attribute>
                  <xsl:value-of select="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Model
Code']/../ix:nominalValue/ix:Text"/>
               </xsl:element>
            </xsl:if>

            <xsl:if test="./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Model Group']/../ix:nominalValue/ix:Text != ''">
               <xsl:element name="model_group">
```

```
                    <xsl:attribute name="fkey"><xsl:value-of select=".”./@id"/></xsl:attribute>
                    <xsl:value-of select=".”./ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Model
Group']/../ix:nominalValue/ix:Text"/>
                </xsl:element>
            </xsl:if>
        </xsl:if>

    </xsl:for-each>
  </xsl:template>

  <!-- ================== -->
  <!-- ================== -->
  <!-- ================== -->
  <!-- ================== -->
  <!-- RELATED SPACE/PROGRAM PROPS -->
  <xsl:template name="EXTpropsSPACEArea">
    <xsl:param name="spaceid"/>
    <xsl:param name="spaceprgid"/>
    <xsl:param name="area"/>
    <xsl:for-each select="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatedObjects/ix:Space[@href
=$spaceid]/../../ix:relatingPropertyDefinition/ix:ElementQuantity">
        <xsl:variable name="elementid" select="@href"/>
        <!-- *If the Placed Space Area Exists, then it overwrites Passed Area. Otherwise Requirements Definition is used -->
        <xsl:if test="/ix:ifcXML/ix:ElementQuantity[@id = $elementid]/ix:quantities/ix:QuantityArea/ix:name[. = 'Area']/../ix:areaValue
&gt; 0">
            <xsl:element name="area">
                <xsl:attribute name="fkey"><xsl:value-of select="$elementid"/></xsl:attribute>
                <xsl:value-of select="/ix:ifcXML/ix:ElementQuantity[@id = $elementid]/ix:quantities/ix:QuantityArea/ix:name[. =
'Area']/../ix:areaValue"/>
            </xsl:element>
        </xsl:if>

        <xsl:if test="not(/ix:ifcXML/ix:ElementQuantity[@id = $elementid]/ix:quantities/ix:QuantityArea/ix:name[. =
'Area']/../ix:areaValue &gt; 0)">
            <xsl:element name="area">
                <xsl:attribute name="fkey"><xsl:value-of select="$spaceid"/></xsl:attribute>
                <xsl:value-of select="$area"/>
            </xsl:element>
        </xsl:if>

    </xsl:for-each>
  </xsl:template>

  <!-- ================== -->
  <!-- ================== -->
  <!-- ================== -->
  <!-- ================== -->
  <!-- CALC FLOORS ROUTINE -->
  <!-- *Counts the number of stories above and below ground. -->
  <xsl:template name="calcfloor">
    <xsl:param name="buildid"/>
    <xsl:variable name="storeyids" select="/ix:ifcXML/ix:RelAggregates/ix:relatingObject/ix:Building[@href =
$buildid]/../../ix:relatedObjects/ix:BuildingStorey/@href"/>

<xsl:if test="count(/ix:ifcXML/ix:BuildingStorey[@id=$storeyids]/ix:elevation[.&gt;= 0]) &gt; 0">
        <xsl:element name="stories_ag">
            <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
            <xsl:value-of select="count(/ix:ifcXML/ix:BuildingStorey[@id=$storeyids]/ix:elevation[.&gt;= 0])"/>
        </xsl:element>
    </xsl:if>

    <xsl:if test="count(/ix:ifcXML/ix:BuildingStorey[@id=$storeyids]/ix:elevation[.&lt; 0]) &gt; 0">
        <xsl:element name="stories_bg">
            <xsl:attribute name="fkey"><xsl:value-of select="@id"/></xsl:attribute>
            <xsl:value-of select="count(/ix:ifcXML/ix:BuildingStorey[@id=$storeyids]/ix:elevation[.&lt; 0])"/>
        </xsl:element>
    </xsl:if>
  </xsl:template>

  <!-- ================== -->
  <!-- ================== -->
  <!-- ================== -->
```

```
   <!-- ================= -->
   <!-- CALC STAIR/ELEVATOR ROUTINE -->
   <!-- *Count the number of Stairwells and Elevator Shafts. -->
   <xsl:template name="calcstrelv">
      <xsl:param name="buildid"/>
      <xsl:param name="storeyids"/>
      <xsl:variable name="storeycount" select="count($storeyids)"/>
      <xsl:variable name="spaceids" select="/ix:ifcXML/ix:RelAggregates/ix:relatingObject/ix:BuildingStorey[@href = $storey-
ids]/../../ix:relatedObjects/ix:Space/@href"/>
      <xsl:variable name="psetids" select="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatedObjects/ix:Space[@href
=$spaceids]/../../ix:relatingPropertyDefinition/ix:PropertySet/@href"/>
      <xsl:variable name="elvcount" select="count(/ix:ifcXML/ix:PropertySet[@id =
$psetids]/ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text[contains(.,'elevator') or
contains(.,'Elevator')or contains(.,'ELEVATOR') or .='ELV' or .='MEV'])"/>
      <xsl:variable name="strcount" select="count(/ix:ifcXML/ix:PropertySet[@id =
$psetids]/ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'Code']/../ix:nominalValue/ix:Text[contains(.,'stair') or contains(.,'Stair')
or contains(.,'STAIR') or .='STR' or .='MSR'])"/>
      <xsl:if test="$storeycount &gt; 0">
         <xsl:if test="$elvcount &gt; 0">
            <xsl:element name="elevators">
               <xsl:value-of select="$elvcount div $storeycount"/>
            </xsl:element>
         </xsl:if>
         <xsl:if test="$strcount &gt; 0">
            <xsl:element name="stairwells">
               <xsl:value-of select="$strcount div $storeycount"/>
            </xsl:element>
         </xsl:if>
      </xsl:if>
   </xsl:template>
   <!-- ================= -->
   <!-- ================= -->
   <!-- ================= -->
   <!-- ================= -->
   <!-- SITEWORKS ROUTINE -->
   <!-- *Create a Sitework Container for Each Site -->
   <xsl:template name="sitework">
      <xsl:param name="projid"/>
      <xsl:param name="siteids"/>

      <xsl:for-each select="$siteids">
         <xsl:variable name="siteid" select="."/>
         <xsl:variable name="sitename" select="/ix:ifcXML/ix:Site[@id = $siteid]/ix:name"/>
         <xsl:element name="sitework">
            <xsl:element name="name">
               <xsl:value-of select="$sitename"/>
            </xsl:element>
            <xsl:element name="models">
               <xsl:call-template name="sitework2">
                  <xsl:with-param name="siteid" select="$siteid"/>
               </xsl:call-template>
            </xsl:element>
         </xsl:element>
      </xsl:for-each>
   </xsl:template>

   <!-- *Determines if there are any PACES_MODEL or Sitework properties associates with the site. -->
   <xsl:template name="sitework2">
      <xsl:param name="siteid"/>
      <xsl:variable name="psetids" select="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatedObjects/ix:Site[@href
=$siteid]/../../ix:relatingPropertyDefinition/ix:PropertySet/@href"/>
      <!-- *Important!  -->
      <!-- *ADD Sitework Models here: New Sitework Models need to be added here in the period "." delimited string -->
      <xsl:variable name="techs" select="concat('UndergroundElectricalDistribution.ParkingLots.Fencing.WaterDistribution', '.')"/>
      <!-- *Important!  -->

      <xsl:for-each select="/ix:ifcXML/ix:PropertySet[@id = $psetids]/ix:name[contains($techs, translate(concat(., '.'),' ', ''))]">
         <xsl:element name="model">
            <xsl:element name="sw_code">
               <xsl:value-of select="."/>
            </xsl:element>
            <xsl:for-each select="../../ix:hasProperties/ix:PropertySingleValue">
```

```
                <xsl:element name="sw_parm">
                  <xsl:element name="name">
                    <xsl:value-of select=."/ix:name"/>
                  </xsl:element>
                  <xsl:element name="value">
                    <xsl:value-of select=."/ix:nominalValue/*"/>
                  </xsl:element>
                </xsl:element>
              </xsl:for-each>
            </xsl:element>
          </xsl:for-each>
        </xsl:template>
        <!-- ================= -->
        <!-- ================= -->
        <!-- ================= -->
        <!-- ================= -->
        <!-- StoreyHeight-->
        <xsl:template name="StoreyHeights">
          <!-- *PACES has a single Floor Height property where BC can have a height for every Storey. We only grab the first
occourance and use it in PACES. -->
          <xsl:param name="buildid"/>
          <xsl:param name="storeyidsag"/>
          <xsl:param name="storeyidsbg"/>

          <xsl:variable name="psetidsag" select="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatedObjects/ix:BuildingStorey[@href =
$storeyidsag]/../../ix:relatingPropertyDefinition/ix:PropertySet/@href"/>
          <xsl:variable name="psetidsbg" select="/ix:ifcXML/ix:RelDefinesByProperties/ix:relatedObjects/ix:BuildingStorey[@href =
$storeyidsbg]/../../ix:relatingPropertyDefinition/ix:PropertySet/@href"/>

          <xsl:if test="/ix:ifcXML/ix:PropertySet[@id = $psetidsag]/ix:name[. = 'BCSys-
tem']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'FloorToFloorHeight']/../ix:nominalValue/ix:Real != """>
            <xsl:element name="floor2floor_height_ag">
              <xsl:attribute name="fkey"><xsl:value-of select=."/@id"/></xsl:attribute>
              <xsl:value-of select="/ix:ifcXML/ix:PropertySet[@id = $psetidsag]/ix:name[. = 'BCSys-
tem']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'FloorToFloorHeight']/../ix:nominalValue/ix:Real"/>
            </xsl:element>
          </xsl:if>

          <xsl:if test="/ix:ifcXML/ix:PropertySet[@id = $psetidsbg]/ix:name[. = 'BCSys-
tem']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'FloorToFloorHeight']/../ix:nominalValue/ix:Real != """>
            <xsl:element name="floor2floor_height_bg">
              <xsl:attribute name="fkey"><xsl:value-of select=."/@id"/></xsl:attribute>
              <xsl:value-of select="/ix:ifcXML/ix:PropertySet[@id = $psetidsbg]/ix:name[. = 'BCSys-
tem']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'FloorToFloorHeight']/../ix:nominalValue/ix:Real"/>
            </xsl:element>
          </xsl:if>

          <xsl:if test="/ix:ifcXML/ix:PropertySet[@id = $psetidsag]/ix:name[. = 'BCSys-
tem']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'FloorToCeilingHeight']/../ix:nominalValue/ix:Real != """>
            <xsl:element name="floor2ceiling_height_ag">
              <xsl:attribute name="fkey"><xsl:value-of select=."/@id"/></xsl:attribute>
              <xsl:value-of select="/ix:ifcXML/ix:PropertySet[@id = $psetidsag]/ix:name[. = 'BCSys-
tem']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'FloorToCeilingHeight']/../ix:nominalValue/ix:Real"/>
            </xsl:element>
          </xsl:if>

          <xsl:if test="/ix:ifcXML/ix:PropertySet[@id = $psetidsbg]/ix:name[. = 'BCSys-
tem']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'FloorToCeilingHeight']/../ix:nominalValue/ix:Real != """>
            <xsl:element name="floor2ceiling_height_bg">
              <xsl:attribute name="fkey"><xsl:value-of select=."/@id"/></xsl:attribute>
              <xsl:value-of select="/ix:ifcXML/ix:PropertySet[@id = $psetidsbg]/ix:name[. = 'BCSys-
tem']/../ix:hasProperties/ix:PropertySingleValue/ix:name[. = 'FloorToCeilingHeight']/../ix:nominalValue/ix:Real"/>
            </xsl:element>
          </xsl:if>

        </xsl:template>

        <!-- END STYLESHEET -->
</xsl:stylesheet>
<!-- SHA-1 Checksum (Remove this line prior to re-validation): 9969C0096C130D8A3CEC074F4D0E9F8E3E627428  -->
```

Final XML Schema File

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by Sean Martin (Earth Tech) -->
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Dan Murphy (Dan Murphy) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDe-
fault="unqualified" version="0.5">
   <xs:element name="earthtech_paces">
      <xs:annotation>
         <xs:documentation>Root of PACES XML file</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element name="project" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:all>
                     <xs:element name="agency" nillable="false" minOccurs="0">
                        <xs:complexType>
                           <xs:simpleContent>
                              <xs:extension base="xs:string">
                                 <xs:attribute name="id" type="xs:ID" use="optional"/>
                                 <xs:attribute name="fkey" type="xs:string" use="optional"/>
                              </xs:extension>
                           </xs:simpleContent>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="date_prepared" nillable="false">
                        <xs:complexType>
                           <xs:simpleContent>
                              <xs:extension base="xs:string">
                                 <xs:attribute name="id" type="xs:ID" use="optional"/>
                                 <xs:attribute name="fkey" type="xs:string" use="optional"/>
                              </xs:extension>
                           </xs:simpleContent>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="facilities">
                        <xs:complexType>
                           <xs:choice>
                              <xs:element name="facility" maxOccurs="unbounded">
                                 <xs:complexType>
                                    <xs:all>
                                       <xs:element name="name" nillable="false">
                                          <xs:complexType>
                                             <xs:simpleContent>
                                                <xs:extension base="xs:string">
                                                   <xs:attribute name="id" type="xs:ID" use="optional"/>
                                                   <xs:attribute name="fkey" type="xs:string" use="optional"/>
                                                </xs:extension>
                                             </xs:simpleContent>
                                          </xs:complexType>
                                       </xs:element>
                                       <xs:element name="model_code" nillable="false">
                                          <xs:complexType>
                                             <xs:simpleContent>
                                                <xs:extension base="xs:string">
                                                   <xs:attribute name="id" type="xs:ID" use="optional"/>
                                                   <xs:attribute name="fkey" type="xs:string" use="optional"/>
                                                </xs:extension>
                                             </xs:simpleContent>
                                          </xs:complexType>
                                       </xs:element>
                                       <xs:element name="gross_area" nillable="false">
                                          <xs:complexType>
                                             <xs:simpleContent>
                                                <xs:extension base="xs:double">
                                                   <xs:attribute name="id" type="xs:ID" use="optional"/>
```

```
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="spaces">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="space" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
              <xs:all>
                <xs:element name="area" nillable="false">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:double">
                        <xs:attribute name="id" type="xs:ID" use="optional"/>
                        <xs:attribute name="fkey" type="xs:string" use="optional"/>
                        <xs:attribute name="units" type="xs:string" use="optional"/>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="code" nillable="false">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:string">
                        <xs:attribute name="id" type="xs:ID" use="optional"/>
                        <xs:attribute name="fkey" type="xs:string" use="optional"/>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="door_density" nillable="false" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:integer">
                        <xs:attribute name="id" type="xs:ID" use="optional"/>
                        <xs:attribute name="fkey" type="xs:string" use="optional"/>
                        <xs:attribute name="units" type="xs:string" use="optional"/>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="partitions" nillable="false" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:double">
                        <xs:attribute name="id" type="xs:ID" use="optional"/>
                        <xs:attribute name="fkey" type="xs:string" use="optional"/>
                        <xs:attribute name="units" type="xs:string" use="optional"/>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="plumbing_fixtures" nillable="false" minOccurs="0">
                  <xs:complexType>
                    <xs:simpleContent>
                      <xs:extension base="xs:integer">
                        <xs:attribute name="id" type="xs:ID" use="optional"/>
                        <xs:attribute name="fkey" type="xs:string" use="optional"/>
                        <xs:attribute name="units" type="xs:string" use="optional"/>
                      </xs:extension>
                    </xs:simpleContent>
                  </xs:complexType>
                </xs:element>
                <xs:element name="special_doors" nillable="false" minOccurs="0">
                  <xs:complexType>
```

```
                        <xs:simpleContent>
                          <xs:extension base="xs:double">
                            <xs:attribute name="id" type="xs:ID" use="optional"/>
                            <xs:attribute name="fkey" type="xs:string" use="optional"/>
                            <xs:attribute name="units" type="xs:string" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="wall_finish" nillable="false" minOccurs="0">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:double">
                            <xs:attribute name="id" type="xs:ID" use="optional"/>
                            <xs:attribute name="fkey" type="xs:string" use="optional"/>
                            <xs:attribute name="units" type="xs:string" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                  </xs:all>
                  <xs:attribute name="id" type="xs:ID" use="optional"/>
                  <xs:attribute name="fkey" type="xs:string" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="walls">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="wall" minOccurs="0" maxOccurs="3">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="backup" nillable="false">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute name="id" type="xs:ID" use="optional"/>
                            <xs:attribute name="fkey" type="xs:string" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="percent" nillable="false">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute name="id" type="xs:ID" use="optional"/>
                            <xs:attribute name="fkey" type="xs:string" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="type" nillable="false">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute name="id" type="xs:ID" use="optional"/>
                            <xs:attribute name="fkey" type="xs:string" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                  </xs:all>
                  <xs:attribute name="id" type="xs:ID" use="optional"/>
```

```
                <xs:attribute name="fkey" type="xs:string" use="optional"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="bay_type" nillable="false" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id" type="xs:ID" use="optional"/>
              <xs:attribute name="fkey" type="xs:string" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="cooling_load" nillable="true" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id" type="xs:ID" use="optional"/>
              <xs:attribute name="fkey" type="xs:string" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="cooling_sys" nillable="false" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id" type="xs:ID" use="optional"/>
              <xs:attribute name="fkey" type="xs:string" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="date_prepared" nillable="false" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:date">
              <xs:attribute name="id" type="xs:ID" use="optional"/>
              <xs:attribute name="fkey" type="xs:string" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="electrical_load" nillable="false" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id" type="xs:ID" use="optional"/>
              <xs:attribute name="fkey" type="xs:string" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="elevators" nillable="false" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:integer">
              <xs:attribute name="id" type="xs:ID" use="optional"/>
              <xs:attribute name="fkey" type="xs:string" use="optional"/>
              <xs:attribute name="units" type="xs:string" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
```

```
</xs:element>
<xs:element name="ext_doors" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
        <xs:attribute name="units" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="ext_wall_area" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:double">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
        <xs:attribute name="units" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="ext_window_area" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:double">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
        <xs:attribute name="units" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="cat_code" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="comments" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="description" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="model_group" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
```

```
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="stories_ag" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:integer">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="stories_bg" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:integer">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="type" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="floor2ceiling_height_ag" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:double">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="floor2ceiling_height_bg" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:double">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="floor2floor_height_ag" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:double">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
```

```
</xs:element>
<xs:element name="floor2floor_height_bg" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:double">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
        <xs:attribute name="units" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="floor_structure" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="footprint" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:double">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
        <xs:attribute name="units" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="foundation_type" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="heat_gen_sys" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="heating_load" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="fkey" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="perimeter" nillable="false" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:double">
        <xs:attribute name="id" type="xs:ID" use="optional"/>
```

```
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="plumb_domestic_water" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:integer">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="plumb_sanitary_waste" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:integer">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="plumb_special_sys" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:integer">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="plumbing" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:integer">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="roof_area" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:double">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
            <xs:attribute name="units" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="roof_structure" nillable="false" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
          </xs:extension>
```

```
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="roofing_type" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="soil_type" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="special_doors" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:double">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
          <xs:attribute name="units" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="stair_type" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="stairwells" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:integer">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
          <xs:attribute name="units" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="use" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="pavements" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
```

```
                        <xs:extension base="xs:string">
                          <xs:attribute name="id" type="xs:ID" use="optional"/>
                          <xs:attribute name="fkey" type="xs:string" use="optional"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="facility_percentage" nillable="false" minOccurs="0">
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="xs:string">
                          <xs:attribute name="id" type="xs:ID" use="optional"/>
                          <xs:attribute name="fkey" type="xs:string" use="optional"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="site_improvements" nillable="false" minOccurs="0">
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="xs:string">
                          <xs:attribute name="id" type="xs:ID" use="optional"/>
                          <xs:attribute name="fkey" type="xs:string" use="optional"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="utilities" nillable="false" minOccurs="0">
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="xs:string">
                          <xs:attribute name="id" type="xs:ID" use="optional"/>
                          <xs:attribute name="fkey" type="xs:string" use="optional"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                </xs:all>
                <xs:attribute name="id" type="xs:ID" use="optional"/>
                <xs:attribute name="fkey" type="xs:string" use="optional"/>
              </xs:complexType>
            </xs:element>
          </xs:choice>
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="markups">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="markup" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:double">
                    <xs:attribute name="id" type="xs:ID" use="optional"/>
                    <xs:attribute name="fkey" type="xs:string" use="optional"/>
                    <xs:attribute name="type" type="xs:string" use="required"/>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="prepared_by" nillable="false">
        <xs:complexType>
```

```
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="program_year" nillable="false">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:anySimpleType">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="name" nillable="false">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="number" nillable="false">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="location" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="comments" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="description" nillable="false" minOccurs="0">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="fkey" type="xs:string" use="optional"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
```

```
<xs:element name="units" nillable="false" minOccurs="0">
   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:string">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
</xs:element>
<xs:element name="region" nillable="false" minOccurs="0">
   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:string">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
</xs:element>
<xs:element name="zip_code" nillable="false" minOccurs="0">
   <xs:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:anySimpleType">
            <xs:attribute name="id" type="xs:ID" use="optional"/>
            <xs:attribute name="fkey" type="xs:string" use="optional"/>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
</xs:element>
<xs:element name="siteworks">
   <xs:complexType>
      <xs:choice>
         <xs:element name="sitework" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
               <xs:all>
                  <xs:element name="models">
                     <xs:complexType>
                        <xs:choice>
                           <xs:element name="model" minOccurs="0" maxOccurs="unbounded">
                              <xs:complexType>
                                 <xs:sequence>
                                    <xs:element name="sw_code" nillable="false">
                                       <xs:complexType>
                                          <xs:simpleContent>
                                             <xs:extension base="xs:string">
                                                <xs:attribute name="id" type="xs:ID" use="optional"/>
                                                <xs:attribute name="fkey" type="xs:string" use="optional"/>
                                             </xs:extension>
                                          </xs:simpleContent>
                                       </xs:complexType>
                                    </xs:element>
                                    <xs:element name="sw_parm" maxOccurs="unbounded">
                                       <xs:complexType>
                                          <xs:all>
                                             <xs:element name="name" nillable="false">
                                                <xs:complexType>
                                                   <xs:simpleContent>
                                                      <xs:extension base="xs:string">
                                                         <xs:attribute name="id" type="xs:ID" use="optional"/>
                                                         <xs:attribute name="fkey" type="xs:string" use="optional"/>
                                                      </xs:extension>
                                                   </xs:simpleContent>
                                                </xs:complexType>
                                             </xs:element>
                                             <xs:element name="value" nillable="false">
                                                <xs:complexType>
                                                   <xs:simpleContent>
```

```
                                                <xs:extension base="xs:string">
                                                   <xs:attribute name="id" type="xs:ID" use="optional"/>
                                                   <xs:attribute name="fkey" type="xs:string" use="optional"/>
                                                </xs:extension>
                                             </xs:simpleContent>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:all>
                                 </xs:complexType>
                              </xs:element>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name="name" nillable="false">
               <xs:complexType>
                  <xs:simpleContent>
                     <xs:extension base="xs:string">
                        <xs:attribute name="id" type="xs:ID" use="optional"/>
                        <xs:attribute name="fkey" type="xs:string" use="optional"/>
                     </xs:extension>
                  </xs:simpleContent>
               </xs:complexType>
            </xs:element>
         </xs:all>
      </xs:complexType>
   </xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:all>
<xs:attribute name="id" type="xs:ID" use="optional"/>
<xs:attribute name="fkey" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="foreign_file" nillable="false" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="id" type="xs:ID" use="optional"/>
<xs:attribute name="fkey" type="xs:string" use="optional"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Appendix B: PACES Interface Specification Document

PACES Interface

Specification Document

Task 1

XML Interface between PACES and Criteria Composer

Contract DACA42-01-P-0037

Revision 1

February 21, 2001

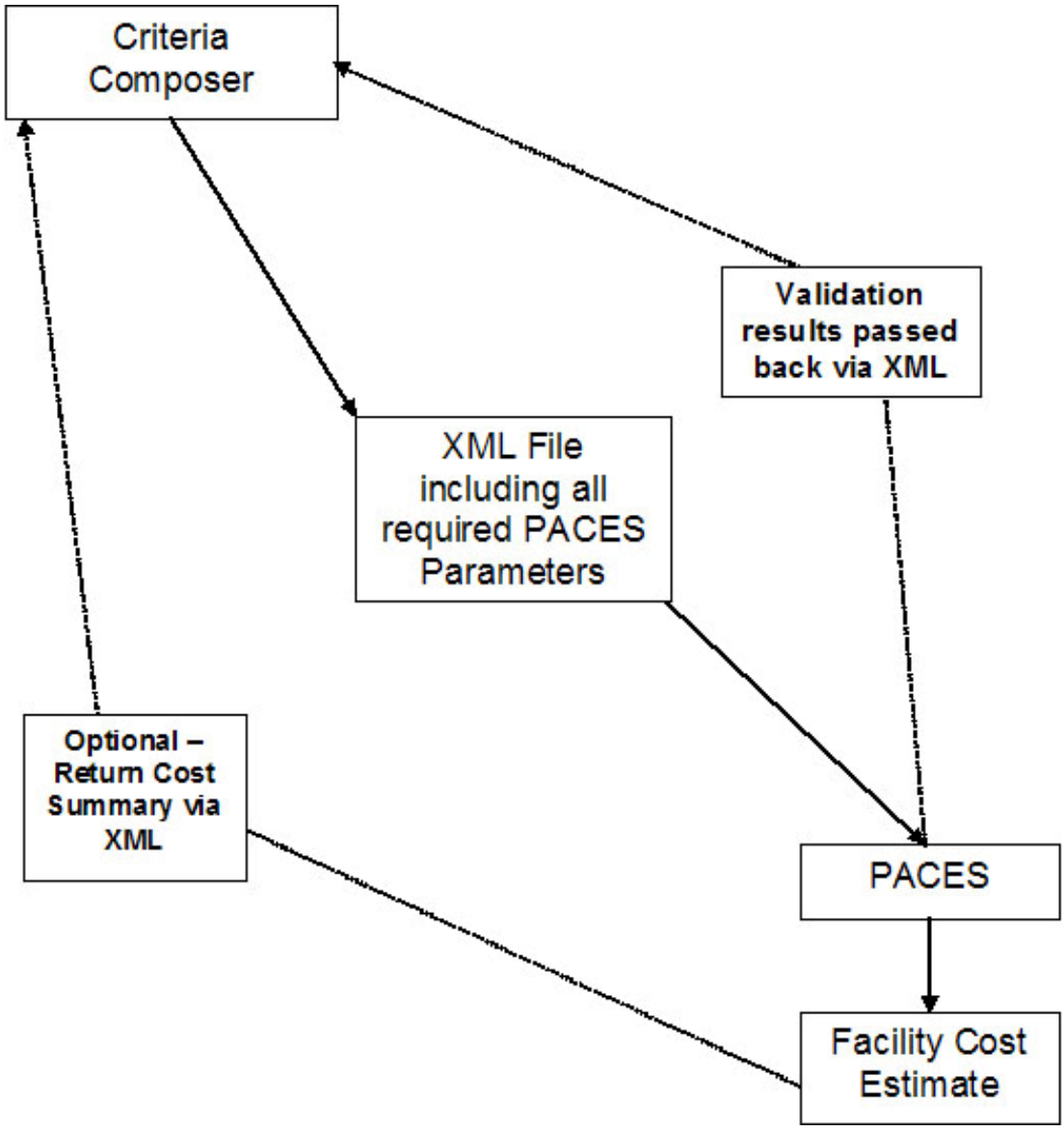
Submitted to ERDC-CERL

by

Talisman Partners, Ltd.

Introduction

This document will define the data requirements for the interface between PACES and Criteria Composer (CC). Task 1 of the Statement of Work states that Talisman [now Earth Tech] will develop a detailed description of PACES data structures that will allow other software developers to map the data structures of other programs to the PACES requirements. Included in the document are descriptions of different "scenarios," or levels of detail, which may be passed into PACES. There are minimum data requirements for each given scenario. This document will describe alternative data sets for the interface between CC and PACES.



Scenarios

This section describes the different "scenarios" of data combinations that may be passed to PACES from CC. PACES will always validate the XML file to ensure the appropriate amount of data is available to generate a cost estimate. The exact building location (as described below) must always be identified. Several different "Facility Parameter" scenarios are offered below. At least one of these must be satisfied before an XML file will be successfully validated and a cost estimate generated.

Location Selection

The user must identify the location (State/Country and City) of the PACES Estimate. This allows the system to use Area Cost Factors (ACFs) specifically developed for each location in the PACES database to adjust the "US Average" costs stored in the system. There are separate material, labor, and equipment factors (by CSI Divisions 01-16) for over 2,000 locations in the PACES system.

The CC XML file must contain either a State (or Country) Code and City Name, or a 5-character "LocationCode." The scenarios for determining the proper location will be as follows:

The State is passed, so the system will default to the "State Average." The PACES system has an "average" location specified for each State and Country.

The State and City are passed. The system will search the PACES Location table to find the appropriate "State/City" combination. If the City is not found, the "State Average" will be used.

If the "LocationCode" is passed, this will be validated and used. This code explicitly states the State/Country and City.

Facility Parameter Scenarios

1 - Facility Parameters "ModelCode" and "GFA"

This is the minimum amount of information to submit to PACES. The ModelCode identifies the exact parametric model within the PACES System, and the GFA (Gross Floor Area) defines the size of the building (SF or m^2). All other parameters can be derived from three pieces of information:

Location (stated above), Model, and Size (GFA). Due to the nature of CC, it is not anticipated that this scenario will be used very often, if at all.

2 - Facility Parameters "ModelGroup" and all "FSACode" and "FSAArea" Parameters

The "ModelGroup" parameter can be used to establish default values for appropriate Facility Quantity and Descriptive Parameters. The Facility GFA value can be calculated as the sum of all Functional Space Area (FSA) sizes (sum of all FSAArea values). Additional Facility Quantity parameters can be derived from this calculated GFA value. The other FSA-specific parameters can be derived from the FSACode and FSAArea values. Talisman will develop the default data to support the ModelGroup concept.

3 - Facility Quantity / Descriptive Parameters, and "FSACode" / "FSAArea" Parameters

If the "ModelGroup" or "ModelCode" parameters are not passed, all appropriate Quantity and Descriptive Parameters may be passed. A Generic Model Group will be established to account for all default Quantity and Descriptive Parameters where they are not passed. All interior FSA parameters (FSACode and FSAArea) must be passed. The sum of all FSAArea values will determine the Facility GFA, and appropriate Quantity Parameters will be derived as necessary.

Additional Parameters

If any of the requirements described above for Facility Parameters are met, any additional parameter (Stories Above Grade, for example) may also be passed to PACES. In addition, any of the "Density" Parameters defined for each FSA [FSAIDD (Number of Interior Doors), for example] may also be passed.

Conflict Resolution

There may be situations where conflicting bits of data are submitted to PACES. If a LocationCode, State and City are all passed, the valid LocationCode would take precedence. If a GFA value is submitted along with FSAArea values for all interior spaces, the sum of the FSAArea values will override the GFA value. This will allow the user to create a "partial estimate," when not all interior space information is known. All normal validation rules will be enforced concerning the Facility Quantity Parameters. Every attempt will be made to prevent erroneous values from being used

in the PACES Building Model program (for example, a 100,000 SF, 1 story building with a footprint of 1,000 SF).

Remaining Sections of this Document

The next section describes each field that may be passed into PACES. All "required" fields are identified. The data type (text, integer, long integer, or single) of each field is listed, along with a Unit of Measure (where applicable), description, and any range or validation rules. This listing divides the data into five separate areas:

1. Project Data
2. Facility Data
3. Facility Quantity Parameters
4. Facility Descriptive Parameters
5. FSA Parameters

Where applicable, fields are related to a specific "List" for validation purposes. Each referenced "List" is displayed in the final section of this document. The lists are as follows:

List	Description
State	States/Countries
Location	Locations used in PACES (partial list)
ModelGroup	Groups of Building Models in PACES
Model	List of all Building Models in PACES
SOT	Soil Type Descriptive Parameter
FST	Floor Structure Type Descriptive Parameter
RST	Roof Structure Type Descriptive Parameter
BAY	Span Length/Bay Size Descriptive Parameter
STT	Stair Type Descriptive Parameter
RGT	Roofing Type Descriptive Parameter
EWT	Exterior Wall Type Descriptive Parameter
EWU	Exterior Wall Backup Descriptive Parameter
HGS	Heat Generating System Descriptive Parameter
CGS	Cooling Generating System Descriptive Parameter
FSA	Functional Space Areas

PACES Building Model Project/Facility Data Fields

Field	Scenario 1	Scenario 2	Scenario 3	Data Type	U/M	Description / Comments	Range/Validation
Project Data Fields							
ProjectName	*	*	*	Text 50		Project Name	
Description				Text 100		Project Description	
State	*			Text 2		State/Country Code	State List
City	*	*		Text 30		City Name	
LocationCode			*	Text 5		5-char T207 Key	Location List
UnitsOption	*	*	*	Text 1		E - English, M – Metric	E / M
Service				Text 23		"Army" – used within PACES to establish Category Code table – N/A here	
ProjectNumber	*	*	*	Text 20		Project Number	
PreparedBy	*	*	*	Text 30		Preparer's Name	
PreparedDate	*	*	*	Date		mm/dd/yyyy format	
ProgramYear	*	*	*	Integer		yyyy format	
ProjectComments				Text 100		Project Comments	
Facility Data Fields							
FacName	*	*	*	Text 50		Facility Name	
FacDescription				Text 255		Facility Description	
FacComments				Text 255		Facility Comments	
ModelGroup	*			Text 3		Paces Model Category	ModelGroup List
ModelCode		*		Text 3		Paces Model Code	Model List
GFA	*			LongInt	SF	Gross Floor Area	200 – 1,000,000
SAG				Integer		Stories Above Grade	0 – 9
SBG				Integer		Stories Below Grade	0 – 2
						*note: SAG + SBG must be > 0	

Facility Quantity Parameters

Field	1	2	3	Data Type	U/M	Description / Comments	Range/Validation
FTP			*	LongInt	SF	Building Footprint	200 to (GFA / (SAG+SBG))
PER			*	LongInt	LF	Building Perimeter	50 – 10,000
RFA			*	LongInt	SF	Roof Area	
FFA			*	Single	FT	Floor to Floor Height Above Grade (average of all floors)	1 – 50
FFB			*	Single	FT	Floor to Floor Height Below Grade (average of all floors)	1 – 50
EWL			*	LongInt	SF	Exterior Wall Area	> 1
EWI			*	LongInt	SF	Exterior Window Area	
EDD			*	LongInt	EA	Exterior Doors	1 – 999
EOD			*	LongInt	SF	Exterior Overhead and Special Doors	0 – 99,999
FCA			*	Single	FT	Floor to Ceiling Height Above Grade (average of all floors)	1 – 50
FCB			*	Single	FT	Floor to Ceiling Height Below Grade (average of all floors)	1 – 50
NSW			*	LongInt	EA	Number of Stairwells	0 – 50
ELV			*	LongInt	EA	Number of Elevators	0 – 50
PDW			*	LongInt	EA	Plumbing Domestic Water Supply	0 – 5,000
PSW			*	LongInt	EA	Plumbing Sanitary Waste System	0 – 5,000
PSS			*	LongInt	EA	Plumbing Special Systems	0 – 5,000
PEQ			*	LongInt	EA	Plumbing Equipment	0 – 5,000
ELD			*	LongInt	AMPS	Electric Load	0 – 50,000
HLD			*	LongInt	MBH	Heating Load	0 – 727,273
CLD			*	Single	TON	Cooling Load	0 – 71,055

Facility Descriptive Parameters

Field	1	2	3	Data Type	U/M	Description / Comments	Range/Validation
SOT			*	LongInt		Soil Type	SOT List
FST			*	LongInt		Floor Structure Type	FST List
RST			*	LongInt		Roof Structure Type	RST List
BAY			*	LongInt		Bay Span / Span Length	BAY List
STT			*	LongInt		Stair Type	STT List
RGT			*	LongInt		Roofing Type	RGT List
EWT			*	LongInt		Exterior Wall Type	EWT List
EWU			*	LongInt		Exterior Wall Backup	EWU List
HGS			*	LongInt		Heat Generating System	HGS List
CGS			*	LongInt		Cooling Generating System	CGS List
Functional Space Area Parameters (repeated for each FSA)							
FSACode		*	*	Text 3		FSA Code	FSA List
FSAArea		*	*	LongInt	SF	FSA Area *note: FSACode and FSAArea may be derived if specific Model (ModelCode) is given	
FSAIDD				LongInt	EA	Interior Doors	
FSAIWD				LongInt	SF	Interior Windows	
FSAIOD				LongInt	SF	Interior Overhead and Special Doors	
FSAIPD				LongInt	SF	Interior Partitions	
FSAPFD				LongInt	EA	Interior Plumbing Fixtures	

Note: Metric equivalents will be used when the "Metric" option is selected:

English	Metric
SF	m2
LF	m
MBH	kW
TON	kW

"*" for Scenario 1/2/3 indicates a required field for that scenario

PACES lists

State List (State and Country)

STATE_ABBR	STATE
AK	ALASKA
AL	ALABAMA
AR	ARKANSAS
AZ	ARIZONA
CA	CALIFORNIA
CO	COLORADO
CT	CONNECTICUT
DC	DISTRICT OF COLUMBIA
DE	DELAWARE
FL	FLORIDA
GA	GEORGIA
HI	HAWAII
IA	IOWA
ID	IDAHO
IL	ILLINOIS
IN	INDIANA
KS	KANSAS
KY	KENTUCKY
LA	LOUISIANA
MA	MASSACHUSETTS
MD	MARYLAND
ME	MAINE
MI	MICHIGAN
MN	MINNESOTA
MO	MISSOURI
MS	MISSISSIPPI
MT	MONTANA
NC	NORTH CAROLINA
ND	NORTH DAKOTA
NE	NEBRASKA
NH	NEW HAMPSHIRE
NJ	NEW JERSEY
NM	NEW MEXICO
NV	NEVADA
NY	NEW YORK
OH	OHIO
OK	OKLAHOMA
OR	OREGON

STATE_ABBR	STATE
PA	PENNSYLVANIA
RI	RHODE ISLAND
SC	SOUTH CAROLINA
SD	SOUTH DAKOTA
TN	TENNESSEE
TX	TEXAS
UT	UTAH
VA	VIRGINIA
VT	VERMONT
WA	WASHINGTON
WI	WISCONSIN
WV	WEST VIRGINIA
WY	WYOMING
AC	ASCENSION
AN	ANTIGUA
AN	ANTILLES
AO	AZORES
AU	AUSTRALIA
BA	BAHAMAS
BD	BERMUDA
BE	BELGIUM
BH	BAHRAIN
CB	CARIBBEAN
CN	CANADA
CR	CRETE
CU	CUBA
DA	DENMARK
DG	DIEGO GARCIA
EG	EGYPT
GE	GERMANY
GL	GREENLAND
GR	GREECE
GU	GUAM
HO	HONDURAS
IC	ICELAND
IT	ITALY
JA	JAPAN
JO	JORDAN
JQ	JOHNSTON ATOLL
KE	KENYA
KO	KOREA

STATE_ABBR	STATE
KU	KUWAIT
KW	KWAJALEIN
LS	LAOS
LU	LUXEMBOURG
MR	MOROCCO
MW	MIDWAY ISLAND
NL	NETHERLANDS
NO	NORWAY
NZ	NEW ZEALAND
OM	OMAN
PC	PANAMA CANAL ZONE
PR	PUERTO RICO
QA	QATAR
RP	PHILIPPINES
SA	SAUDI ARABIA
SE	SEYCHELLES
SF	SOUTH AFRICA
SN	SAIPAN
SO	SOMALI
SP	SPAIN
TH	THAILAND
TQ	KOSRAE ISLAND
TU	TURKEY
TW	TAIWAN
UK	UNITED KINGDOM
US	US
VI	VIRGIN ISLANDS
WK	WAKE ISLAND

Location List (partial list)

LOC_CODE	STATE_ABBR	LOCATION
10945	DE	WILMINGTON
10999	DE	WILMINGTON/NEW CASTLE CO APT
11000	DC	DISTRICT OF COLUMBIA AVG
11031	DC	BOLLING AFB
11040	DC	DOD VARIOUS
11381	DC	HARRY DIAMOND LABORATORY
11532	DC	NAVY HEADQUARTERS
11564	DC	FORT MCNAIR
11609	DC	WASHINGTON NAVY YARD
11688	DC	PENTAGON BUILDING

11850	DC	US SOLDIERS HOME
11933	DC	WALTER REED AMC
11935	DC	WASHINGTON
11936	DC	WASH-BALT DEFENSE AREA
12000	FL	FLORIDA STATE AVERAGE
12001	FL	EGLIN 3
12002	FL	EGLIN 9
12004	FL	FLOR VAR
12005	FL	AVON PAR
12006	FL	AVON RNG
12007	FL	COCOA B1
12008	FL	COCOA B2
12009	FL	COCOA OC
12010	FL	FT PIERCE
12011	FL	JONATHAN
12012	FL	MALABAR
12013	FL	MELBOURN
12014	FL	STUART
12015	FL	WABASSO
12016	FL	EGLIN NR 3
12080	FL	FORT BLANDING
12111	FL	WHITING FIELD
12211	FL	BARRANCAS
12222	FL	CECIL FIELD
12228	FL	EAU GALLIE
12232	FL	EGLIN AFB
12246	FL	LYNN HAVEN
12273	FL	JACKSONVILLE NAS
12279	FL	TYNDALL AFB
12367	FL	HOMESTEAD-MIAMI DEFENSE AREA
12370	FL	HOMESTEAD AFB
12401	FL	HURLBURT FIELD (EGLIN #9)
12430	FL	KEY WEST NAS
12450	FL	KENNEDY SPACE CENTER
12451	FL	CAPE CANAVERAL
12470	FL	LAKELAND
12495	FL	MAYPORT NAVAL STATION
12572	FL	MIAMI
12645	FL	OCALA
12679	FL	ORLANDO
12700	FL	PALATKA
12712	FL	PANAMA CITY

12719	FL	PATRICK AFB
12723	FL	PENSACOLA NAS
12724	FL	PENSACOLA
12802	FL	ST PETERSBURG
12819	FL	MACDILL AFB
12883	FL	TAMPA
12960	FL	WHITING FIELD NAVAL AIR STA.
12994	FL	GULF COAST
12998	FL	PORT CANAVERAL
12999	FL	JACKSONVILLE INTL APT ANG
13000	GA	GEORGIA STATE AVERAGE
13001	GA	LEWIS B
13002	GA	ST SIMON
13010	GA	MOODY AFB
13020	GA	ALBANY
13048	GA	FORT GILLEM
13050	GA	AUGUSTA
13069	GA	ATLANTA NAS
13077	GA	FORT BENNING
13091	GA	COLUMBUS
13158	GA	CHAMBLEE
13265	GA	DUBLIN
13270	GA	DOBBINS AFB
13275	GA	EAST POINT
13355	GA	FORT GORDON
13424	GA	HUNTER ARMY AIR FIELD
13450	GA	KINGS BAY
13476	GA	KINGS BAY MIL OCEAN TERM
13567	GA	FORT MCPHERSON
13600	GA	ATLANTA-MARIETTA
13636	GA	ATLANTA
13641	GA	ROBINS AFB
13755	GA	ROME
13811	GA	FORT STEWART
13882	GA	TIFTON
13921	GA	MACON
13974	GA	ATHENS
13981	GA	FORT VALLEY
13998	GA	ALBANY AREA
13999	GA	SAVANNAH INTL APT

ModelGroup List

MODEL_NBR	MODEL_GROUP
002	MEDICAL FACILITIES GROUP
003	STORAGE FACILITIES GROUP
004	ADMINISTRATIVE FACILITIES GROUP
008	COMMUNICATIONS FACILITIES GROUP
009	BASE SUPPORTING FACILITIES
012	AIRCRAFT OPERATIONS
030	MAINTENANCE HANGARS GROUP
040	MAINTENANCE FACILITIES GROUP
060	DINING FACILITIES
070	DORMITORIES
080	LODGING FACILITIES GROUP
AFS	NAF FACILITIES
H01	MILITARY HOUSING GROUP
LC1	LIGHT CONSTRUCTION BUILDINGS
S00	SCHOLASTIC FACILITIES

Model List

MODEL_NBR	MODEL_NAME
001	GENERAL ADMINISTRATIVE FACILITY
005	TROOP SUBSISTENCE WAREHOUSE
006	STORAGE SHED
007	GENERAL PURPOSE MAGAZINE
009	TELEPHONE EXCHANGE BUILDING
011	CENTRAL PREP/COLD STORAGE FACILITY
012	HAZARDOUS MATERIAL STORAGE
013	WRM STORAGE
014	BASE SUPPLY COMPLEX
015	COVERED STORAGE
016	HIGH BAY WAREHOUSE
021	COMMAND HEADQUARTERS FACILITY
022	DATA PROCESSING CENTER
023	GENERAL PURPOSE OPERATION FACILITY
025	COMMUNICATIONS FACILITY MODEL
027	SATCOM CENTER
031	AIRCRAFT MAINTENANCE HANGAR (HIGH BAY)
032	AIRCRAFT MAINTENANCE HANGAR (MEDIUM BAY) (OLD)
033	CORROSION CONTROL HANGAR (HIGH BAY)
034	FUEL SYSTEM MAINTENANCE HANGAR (MEDIUM BAY)
035	NON-DESTRUCTIVE INSPECTION FACILITY
041	AIRCRAFT AVIONICS SHOP
042	AIRCRAFT ENGINE I & R SHOP

043	BASE CIVIL ENGINEERING COMPLEX
044	DEPOT MACHINE SHOP
046	MISSILE MAINTENANCE WITH CLEAN ROOM
048	MISSILE ASSEMBLY AND MAINTENANCE BUILDING
051	PRECISION MEASURING EQUIPMENT LAB
054	VEHICLE MAINTENANCE FACILITY
061	40 - 250 PERSON DINING FACILITY
062	251 - 650 PERSON DINING FACILITY
063	651 - 1000 PERSON DINING FACILITY
064	1001 - 1500 PERSON DINING FACILITY
071	AF ENLISTED/1 + 1 DORM
072	AF ENLISTED/2 + 2 DORM
076	OFFICER 01 - 02
077	OFFICER 03 - UP
081	TEMPORARY LODGING FACILITY MODEL
086	CHILD CARE CENTER
091	TROOP AID STATION
092	DENTAL CLINIC
093	HEALTH/DENTAL CLINIC
094	TROOP MEDICAL CLINIC
095	COMMUNITY HOSPITAL
096	IN-PATIENT NURSING TOWER (MEDICAL CENTER)
097	OUT-PATIENT MEDICAL CLINIC (MEDICAL CENTER)
098	HOSPITAL/LOGISTICS/ANCILLARIES (MEDICAL CENTER)
099	MEDICAL CENTER
09A	AMBULATORY HEALTH CARE FACILITY
100	HIGH TECH LAB
112	GENERAL PURPOSE OPERATION BUILDING
113	VEHICLE MAINTENENCE SHOP
114	CHILD DEVELOPMENT CENTER
115	ARMY MEDIUM COMPANY OPERATIONS FACILITIES
116	FLIGHT SIMULATOR
117	COLLOCATED CLUB
118	GOLF COURSE CLUB HOUSE
119	GOLF EQUIPMENT STORAGE SHED
121	GOLF COURSE MAINTENANCE BUILDING
125	FAM CAMP SUPPORT FACILITY BUILDING
127	POV CAR WASH
128	OUTDOOR RECREATION OPERATONS FACILITY
130	ARTS & CRAFTS CENTER
131	AUTO SKILLS CENTER
134	CONCESSION STAND/PRESS BOX

137	DOCKMASTER OFFICE
141	RECREATION AREA CABIN
142	RECREATION AREA RECEPTION CABIN
144	GOLF COURSE SNACK BAR/RESTROOM
222	SQUADRON OPERATIONS BUILDING
333	COMMUNITY FIRE STATION
444	ARMY PHYSICAL FITNESS CENTER
451	BARRACKS (48 MAN)
452	SOLDIERS COMMUNITY CENTER BUILDING
453	AIRCRAFT MAINTENANCE HANGAR (MEDIUM BAY)
455	CORROSION CONTROL FACILITY (Medium Bay)
457	YOUTH CENTER
464	BARRACKS (96 MAN)
470	STORAGE IGLOO
888	CONTROL TOWER
H02	2 BEDROOM - JNCO/SNCO/CGO - FREE STANDING UNIT
H03	2 BEDROOM - JNCO/SNCO/CGO - CENTER UNIT
H04	2 BEDROOM - JNCO/SNCO/CGO - END UNIT
H05	3 BEDROOM - JNCO - FREE STANDING UNIT
H06	3 BEDROOM - JNCO - CENTER UNIT
H07	3 BEDROOM - JNCO - END UNIT
H08	3 BEDROOM - SNCO/CGO - FREE STANDING UNIT
H09	3 BEDROOM - SNCO/CGO - CENTER UNIT
H10	3 BEDROOM - SNCO/CGO - END UNIT
H11	3 BEDROOM - FGO - FREE STANDING UNIT
H12	4 BEDROOM - JNCO - FREE STANDING UNIT
H13	4 BEDROOM - JNCO - CENTER UNIT
H14	4 BEDROOM - JNCO - END UNIT
H15	4 BEDROOM - SNCO/CGO - FREE STANDING UNIT
H16	4 BEDROOM - SNCO/CGO - CENTER UNIT
H17	4 BEDROOM - SNCO/CGO - END UNIT
H18	4 BEDROOM - FGO - FREE STANDING UNIT
H19	4 BEDROOM - SGO - FREE STANDING UNIT
H20	4 BEDROOM - GO - FREE STANDING UNIT
H21	5 BEDROOM - JNCO/SNCO/CGO - FREE STANDING UNIT
H22	5 BEDROOM - JNCO/SNCO/CGO - CENTER UNIT
H23	5 BEDROOM - JNCO/SNCO/CGO - END UNIT
S01	ELEMENTARY SCHOOL
S02	ELEMENTARY SCHOOL (MEDIUM GRADE)
S03	ELEMENTARY SCHOOL (HIGH GRADE)
S04	MIDDLE SCHOOL
S05	MIDDLE SCHOOL (MEDIUM GRADE)

S06	MIDDLE SCHOOL (HIGH GRADE)
S07	HIGH SCHOOL
S08	HIGH SCHOOL (MEDIUM GRADE)
S09	HIGH SCHOOL (HIGH GRADE)
VA1	VA OUTPATIENT CLINIC (DIRECT LEASE)
VA2	VA OUTPATIENT CLINIC
VA3	VA NURSING HOME CARE UNIT (NHCU)
VA4	VA DOMICILIARY
VA5	VA PSYCHIATRIC HOSPITAL
VA6	VA CLINIC ADDITION
VA7	VA MEDICAL CENTER

SOT List (Soil Type)

Option	Desc
1	Low Bearing Capacity less than 10000 kg/m2 (2000 PSF)
2	Average Bearing Capacity 10000 - 17000 kg/m2 (2000 - 3500 PSF)
3	High Bearing Capacity greater than 17000 kg/m2 (3500 PSF)

FST List (Floor Structure Type)

Option	Desc
1	Concrete Frame
2	Steel Frame-Light Load w/ Reinforced Concrete Deck
3	Steel Frame-Heavy Load w/ Reinforced Concrete Deck
4	Steel Frame-Light Load-Mtl Joists-Stl Deck w/ Conc Fill
5	Steel Frame-Heavy Load-Mtl Joists-Stl Deck w/ Conc Fill
6	Load Bearing Walls-Mtl Joists-Stl Deck w/ Conc Fill
7	Load Bearing Walls-Wood Joist Floor System w/ Deck
8	Ld Bearing Walls-Precast/Prestressed w/ Conc Topping
9	N/A
10	Concrete Frame Light Load
21	Wood Joist Floor System W/ Deck

RST List (Roof Structure Type)

Option	Desc
1	Concrete Frame
2	Steel Frame-Light Load w/ Reinforce Concrete Roof Deck
3	Steel Frame-Heavy Load w/ Reinforce Concrete Roof Deck
4	Steel Frame-Light Load-Metal Joists-Steel Roof Deck
5	Steel Frame-Heavy Load-Metal Joists-Steel Roof Deck
6	Load Bearing Walls-Metal Joists-Steel Roof Deck
7	Load Bearing Walls-Wood Truss w/ Ply Roof Deck
8	Load Bearing Walls-Precast/Prestressed Slab w/ Concrete Topping

9	Aircraft Hangar High Bay
10	Rigid Frame (Metal Building)
11	Concrete Frame Light Load
21	2x4 Wood Truss Roof System w/ Deck
22	2x6 Wood Truss Roof System w/ Deck

BAY List (Span Length/Bay Size)

Option	Desc
1	Small 28-56 m2 (0-9 m) (300-600 SF (0-30 LF))
2	Average 56-111 m2 (9-15 m) (600-1200 SF (30-50 LF))
3	Large 111-149 m2 (15-29 m) (1200-1600 SF (50-95 LF))
4	Special 149-279 m2 (29-44 m) (1600-3000 SF (95-144 LF))
5	Hangar High Bay
21	N/A

STT List (Stair Type)

Option	Desc
1	Concrete
2	Metal Pan
3	Wood
4	N/A

RGT List (Roofing Type)

Option	Desc
1	Single Membrane
2	Built Up
3	Shingle
4	Standing Seam Metal
5	Clay Tile
6	Metal (Typical Metal Building)
21	Strip Shingle
22	Architechural Shingle
23	Concrete Tile

EWT List (Exterior Wall Type)

Option	Desc
1	Brick Veneer
2	100mm (4") Split Rib Masonry Veneer
3	200mm (8") Split Rib Masonry
4	200mm (8") Masonry Block
5	Tilt-Up Concrete

6	Exposed Aggregate Precast
7	300mm (12") CIP Conc. w/ Exposed Aggregate Finish
8	Metal Sandwich Panel
9	Metal Siding (Typical Metal Building)
10	Stucco
11	E.I.F.S. (Dryvit)
12	Glazed Curtain Wall
13	Glazed Curtain Wall with Spandrel
14	Cedar Lap Siding
15	Vinyl Siding
16	Aluminum Siding
17	N/A

EWU List (Exterior Wall Backup)

Option	Desc
1	200mm (8") Masonry / Painted
2	200mm (8") Masonry w/Metal Furring Strip/Painted Gypsum Board
3	50mm X 100mm (2 X 4) Wood Stud with Painted Gypsum Board
4	92mm (3-5/8") Metal Stud with Painted Gypsum Board
5	150mm (6") Metal Stud with Painted Gypsum Board
6	Metal Furring Strip w/Painted Gypsum Board
7	Paint back of Veneer
8	No Back-Up
21	Load Bearing 50mm X 100mm (2x4) Wood Stud
22	Load Bearing Masonry

HGS List (Heat Generating System)

Option	Desc
1	Steam Boilers
2	Hot Water Boilers - Oil Fired
3	Hot Water Boilers - Gas Fired
4	Gas Furnace
5	Offsite Central Plant
6	N/A
8	Split System Heat Pump
9	Water Source Heat Pump

CGS List (Cooling Generating System)

Option	Desc
1	Reciprocating - Chiller
2	Reciprocating - Direct Expansion
3	Centrifugal

4	Absorption
5	Offsite Central Plant
6	N/A
7	Residential
8	Split System Heat Pump
9	Water Source Heat Pump

FSA List (Functional Space Areas)

FSA_CD	DESC
03A	COMMAND SUITE - DENTAL CLINIC
03B	COMMAND SUITE - HEALTH/DENTAL CLINIC
03C	COMMAND SUITE - TROOP MEDICAL CLINIC
03D	COMMAND SUITE - IN-PATIENT NURSING TOWER
03E	COMMAND SUITE - COMMUNITY HOSPITAL
03G	COMMAND SUITE - OUT-PATIENT MEDICAL CLINIC
03H	COMMAND SUITE HOSPITAL/LOGISTICS/ANCILLARIES
05	MEDICAL LIBRARY
07D	NURSING ADMINISTRATION - IN-PATIENT NURSING TOWER
07G	NURSING ADMINISTRATION - OUT-PATIENT MED CLINIC
09	COMPTROLLER
096	FIRE STATION VEHICLE APPARATUS BAYS
097	RETAIL ISSUE SPACE
098	ELEVATED RUNNING TRACK
100	VA MS&N NURSING UNITS
102	VA INTENSIVE CARE NURSING UNITS
104	VA SPINAL CORD INJURY UNIT (SCI)
106	VA NURSING HOME CARE UNIT (NHCU)
10C	CORROSION CONTROL BUILDING SUPPORT
11	PERSONNEL
110	VA MENTAL HEALTH & BEHAVIORAL NURS. UNITS
11D	PERSONNEL - INPATIENT NURSING TOWER
11G	PERSONNEL - OUTPATIENT MEDICAL CLINIC
12	INFORMATION MANAGEMENT
13	PLANS, OPERATIONS & TRAINING
14	PATIENT ADMINSTRATION
14D	PATIENT ADMIN - INPATIENT NURSING TOWER
14G	PATIENT ADMIN - OUTPATIENT MEDICAL CLINIC
14H	PATIENT ADMIN - HOSPITAL/LOGISTICS/ANCILLARIES
15	FOOD SERVICE
17A	LOGISTICS - DENTAL CLINIC
17B	LOGISTICS - HEALTH/DENTAL CLINIC
17C	LOGISTICS - TROOP MEDICAL CLINIC

17E	LOGISTICS - COMMUNITY HOSPITAL
17F	LOGISTICS - TROOP AID STATION
17H	LOGISTICS - HOSPITAL/LOGISTICS/ANCILLARIES
17I	LOGISTICS - AMBULATORY HEALTH CARE FACILITY
18C	CLINIC ADMIN SUPPORT - TROOP MEDICAL CLINIC
18G	CLINIC ADMIN SUPPORT - OUT-PATIENT MED CLINIC
19B	PRIMARY CARE CLINIC - HEALTH/DENTAL CLINIC
19C	PRIMARY CARE CLINIC - TROOP MEDICAL CLINIC
19E	PRIMARY CARE CLINIC - COMMUNITY HOSPITAL
19F	PRIMARY CARE CLINIC - TROOP AID STATION
19H	PRIMARY CARE CLINIC - HOSPITAL/LOGISTICS/ANCILLARI
202	VA SUBSTANCE ABUSE CLINIC
204	VA AUDIOLOGY & SPEECH PATHOLOGY
206	VA CANTEEN SERVICE
208	VA CHAPLAIN SERVICE
20C	FLIGHT/UNDERSEA MEDICINE - TROOP MEDICAL CLINIC
20E	FLIGHT/UNDERSEA MEDICINE - COMMUNITY HOSPITAL
20G	FLIGHT/UNDERSEA MEDICINE - OUT-PATIENT CLINIC
21	ALLERGY/IMMUNIZATIONS
210	VA CARDIOVASCULAR LABORATORY
212	VA PULMONARY MEDICINE
214	VA CLINICAL SERVICES ADMINISTRATION
218	VA VETERANS ASSISTANCE UNIT
22	CARDIOVASCULAR/PULMONARY
220	VA CREDIT UNION
222	VA DENTAL SERVICE
224	VA DIETETICS
226	VA ELECTROENCEPHALOGRAPHY LAB (EEG)
23	DERMATOLOGY
230	VA ENGINEERING SERVICE
233	VA EYE CLINIC
234	VA FISCAL SERVICE
238	VA HOSPITAL DIRECTORS SUITE
239	VA INFORMATION RESOURCE MGMT SVC
24	HEMATOLOGY/ONCOLOGY
240	VA LABORATORY SERVICE
244	LOBBY
246	VA MEDICAL ADMINISTRATIVE SERVICE (MAS)
248	VA MEDICAL MEDIA SERVICE
25	NEUROLOGY/RNDO/RHEUMATOLOGY
252	VA NUCLEAR MEDICINE SERVICE
254	VA NURSING SERVICE ADMIN

26	INTERNAL MEDICINE
260	VA MENTAL HEALTH CLINIC
261	VA DAY TREATMENT CENTER
262	VA AMBULATORY CARE
266	VA PERSONNEL
268	VA PHARMACY SERVICE
269	VA RECREATION SERVICE
27	GASTROENTEROLOGY
270	VA REHABILITATION MEDICINE SERVICE
272	VA PSYCHOLOGY SERVICE
274	VA QUARTERS ON-CALL
275	VA MAGNETIC RESONANCE IMAGING (MRI)
276	VA RADIOLOGY SERVICE
277	VA RADIATION THERAPY SERVICE
278	VA MEDICAL RESEARCH & DEVELOPMENT
279	VA POLICE AND SECURITY SERVICE
28	PEDIATRICS
280	VA SERVICE ORGANIZATIONS
282	VA SOCIAL WORK SERVICE
284	VA SUPPLY SERVICE, ADMIN
285	VA SUPPLY SERVICE, SPD
286	VA SURGICAL SERVICE
287	VA DIGESTIVE DISEASES SVC-ENDOSCOPY
29	NEPHROLOGY
290	VA VOLUNTARY SERVICE
291	VA SUPPLY SERVICE, WAREHOUSE
298	VA DATA PROCESSING
30	GENERAL SURGERY
300	VA DAY HOSPITAL
308	VA PROSTHETICS & SENSORY AIDS
31	UROLOGY
312	VA DOMICILIARY
316	VA DIALYSIS
32	NEUROSURGERY
34D	ORTHOPEDICS/PODIATRY - IN-PATIENT NURSING TOWER
34G	ORTHOPEDICS/PODIATRY - OUT-PATIENT MED CLINIC
35	EAR, NOSE, AND THROAT/AUDIOLOGY
36	OPHTHALMOLOGY/OPTOMETRY
38	OBSTETRICS/GYNECOLOGY
40	PREV. MED/OCC THERAPY
400	VA LIBRARY SERVICE
402	VA EDUCATION FACILITIES

406	VA BUILDING MANAGEMENT SERVICE, ADMIN
408	VA LAUNDRY & LINEN OPERATION
40B	PREV. MED/OCC HEALTH - HEALTH/DENTAL CLINIC
40G	PREV. MED/OCC HEALTH - OUTPATIENT MEDICAL CLINIC
41	PSYCHIATRY
410	VA LOCKERS, LOUNGES, TOILETS, SHOWERS (LLTS)
42	PSYCHOLOGY
43	SOCIAL WORK
44	PHYSICAL/OCC THERAPY
44E	PHYSICAL/OCC THERAPY - COMMUNITY HOSPITAL
50C	PHARMACY - TROOP MEDICAL CLINIC
50E	PHARMACY - COMMUNITY HOSPITAL
50F	PHARMACY - TROOP AID STATION
50G	PHARMACY - OUTPATIENT MEDICAL CLINIC
50H	PHARMACY - HOSPITAL/LOGISTICS/ANCILLARIES
51C	RADIOLOGY - TROOP MEDICAL CLINIC
51E	RADIOLOGY - COMMUNITY HOSPITAL
51H	RADIOLOGY - HOSPITAL/LOGISTICS/ANCILLARIES
52	NUCLEAR MEDICINE
53B	PATHOLOGY - HEALTH/DENTAL CLINIC
53C	PATHOLOGY - TROOP MEDICAL CLINIC
53F	PATHOLOGY - TROOP AID STATION
53H	PATHOLOGY - HOSPITAL/LOGISTICS/ANCILLARIES
54	RADIATION THERAPY
57	CENTRAL MATERIEL SERVICES
57E	CENTRAL MATERIEL SERVICES - COMMUNITY HOSPITAL
59	SURGICAL SERVICES
59E	SURGICAL SERVICES - COMMUNITY HOSPITAL
59H	SURGICAL SERVICES - HOSPITAL/LOGISTICS/ANCILLARIES
61	INTENSIVE CARE UNIT
61D	INTENSIVE CARE UNIT - INPATIENT NURSING TOWER
61H	INTENSIVE CARE UNIT - HOSPITAL/LOGISTICS/ANCILLARE
62	CARDIAC CARE UNIT
63D	OBSTETRICAL UNIT - INPATIENT NURSING TOWER
63E	OBSTETRICAL UNIT - COMMUNITY HOSPITAL
63H	OBSTETRICAL UNIT - HOSPITAL/LOGISTICS/ANCILLARIES
65	NURSERY
71D	MEDICAL/SURGICAL UNIT - INPATIENT NURSING TOWER
71E	MEDICAL/SURGICAL UNIT - COMMUNITY HOSPITAL
76	LIGHT CARE UNIT
77	PEDIATRIC UNIT
78	DETOX UNIT

79	PSYCHIATRIC UNIT
83	PATIENT SERVICES
83D	PATIENT SERVICES - IN-PATIENT NURSING TOWER
83E	PATIENT SERVICES - COMMUNITY HOSPITAL
83H	PATIENT SERVICES - HOSPITAL/LOGISTICS/ANCILLARIES
84	CHAPLAIN
85	CLINICAL INVESTIGATE
87	DENTISTRY
88E	DENTAL CLINIC W/DENTAL TUNNEL
91A	CIRCULATION - DENTAL CLINIC
91B	CIRCULATION - HEALTH/DENTAL CLINIC
91F	CIRCULATION - TROOP AID STATION
92A	MECH, ELEC, COMM - DENTAL CLINIC
92B	MECH/ELEC/COMM - HEALTH/DENTAL CLINIC
92E	MECH/ELEC/COMM - COMMUNITY HOSPITAL
92F	MECH, ELEC, COMM - TROOP AID STATION
ABS	ARMY BUILDING SUPPORT
ACL	OFFICE - ARMY CLOSED OFFICE SPACE
ACR	CORRIDOR - SMALL
ADC	ADMIN MODULE - SMALL
ADH	ADMIN MODULE - MEDIUM
ADM	ADMIN MODULE - LARGE
AFC	AIR FORCE CLINICS
AIM	AIRCRAFT MAINTENANCE AREA
AIR	AIR LOCK
AIS	AIS ROOM (AVIONICS INTERMEDIATE STATION)
ALS	ARMY LOCKERS AND SHOWERS
AMC	AUTO SKILLS WELDING/METAL CUTTING
AMS	AUTO SKILLS MACHINE SHOP
ARB	AUTO REPAIR BAY
ARR	AUTO SKILLS PUBLIC RESTROOMS
ASS	AUTO SKILLS STORAGE (SMALL)
AST	ARMY STORAGE
ATR	AUTO SKILLS TOOL ROOM
AUD	AUDITORIUM
AUX	AUXILIARY POWER ROOM
BAQ	BASE SUPPLY STORE INDIVIDUAL EQUIPMENT AND STORAGE
BAS	BATTERY STORAGE ROOM
BCW	BARRACKS COVERED WALKWAYS AND SERVICE AREAS
BEM	BARRACKS MECHANICAL ROOM
BER	BARRACKS ELECTRICAL ROOM
BOX	BOXING ROOM

BQT	BARRACKS QUARTERS /1 + 1
BSO	OPERATIONS BUILDING SUPPORT AREA
BSP	BUILDING SUPPORT AREA
BTR	BARRACKS TELEPHONE EQUIPMENT ROOM
C01	COMMUNITY CENTER ACTIVITES & T.V. ROOM
C02	COMMUNITY CENTER CLOSED OFFICE SPACE
C03	COMMUNITY CENTER LAUNDRY
C04	COMMUNITY CENTER OPEN OFFICE SPACE
C05	COMMUNITY CENTER TOILETS
C06	COMMUNITY CENTER LUNCH ROOM
C07	COMMUNITY CENTER MAIL ROOM
C08	COMMUNITY CENTER TELEPHONE ROOM
C09	COMMUNITY CENTER MECHANICAL ROOM
C10	COMMUNITY CENTER ELECTRICAL ROOM
C11	GENERAL PURPOSE STORAGE
C12	COMMUNITY CENTER BUILDING SUPPORT AREA
CAL	CALIBRATION/REPAIR AREA
CBS	CONTROL TOWER BUILDING SUPPORT AREA
CC7	PAINT BAY
CC8	CORROSION CONTROL PLENUM
CC9	CORROSION CONTROL MECHANICAL ROOM
CCH	HOURLY CARE CENTER
CCR	COOK AND CLEAN ROOMS
CES	COVERED ENTRYWAY (STEEL TRUSS)
CEW	COVERED ENTRY/WALKWAY (BUILT-UP ROOF)
CFP	CHILLERS AND FOOD PREP AREA
CGN	CORRIDOR (GENERAL)
CGS	CONTROL TOWER GENERAL STORAGE
CHD	TODDLER AREA
CLN	CLEAN ROOM
CME	CONTROL TOWER MECHANICAL & ELECTRICAL ROOM
COM	COM. CTR./MESSAGE DIST/BCC OFFICE (SCIF)
CON	COMPRESSOR BALANCE AREA
COO	COOLER
COR	CONTROL TOWER OPERATIONS ROOM
COS	CLOSED OFFICE SPACE (SMALL)
CRR	CONTROL TOWER RESTROOMS
CSD	CONCESSION STAND
CSF	COLD STORAGE FREEZER AREA-SMALL (10'-14' CLG HGT)
CSL	COLD STORAGE FREEZER AREA-LARGE (15'-22' CLG HGT)
CTO	CONTROL TOWER CLOSED OFFICE SPACE
CVE	COVERED AIRCRAFT ENGINE STORAGE

CVN	COVERED EXTERIOR ENTRYWAY
CVR	COVERED EXTERIOR LOADING DOCK
CVT	COVERED EXTERIOR LOADING DOCK W/LEVELER
CVW	COVERED WALKWAYS AND SERVICE AREAS
CWC	CAR WASH CLOSED BAY
CWL	COVERED ENTRY
CWO	CAR WASH OPEN BAY
CWP	CLASSROOM WITH PARTITION
DBA	DINING/ BANQUET AREA
DEP	DEPOT MACHINE SHOP AREA
DET	DEPOT MAINTENANCE ROOM
DFC	SERVERY FOR CLUBS
DMB	EXTERIOR ENTRY MENS/WOMENS RESTROOM
DMC	DOCKMASTER CLOSED OFFICE
DPL	DATA PROCESSING AREA (LARGE)
DPR	DATA PROCESSING ROOM (SMALL)
DPS	DATA SUPPLY ROOM
DSA	DINING AREA; 40-250 PERSON
DSB	DINING AREA; 251-650 PERSON
DSC	DINING AREA; 651-1000 PERSON
DSD	DINING AREA; 1001-1500 PERSON
ENM	ENGINE MAINTENANCE AREA
EQT	TELEPHONE EQUIPMENT ROOM
EWH	ENTRYWAY WITH HANDRAIL
EXP	EXPOSURE ROOM
EXR	EXERCISE ROOM
EXT	EXTERNAL TANK AND FUEL BLADDER MAINTENANCE
FAC	FACILITIES MANAGEMENT
FBS	FIRE STATION BUILDING SUPPORT AREA
FCO	FIRE STATION CLOSED OFFICE SPACE
FDR	FIRE STATION DINING AREA
FER	FIRE STATION EXCERCISE ROOM
FIL	FILM PROCESSING, STORAGE AND LAB
FIN	FOOD INSPECTION LABORATORY
FLG	FIRE STATION LOUNGE/GAMEROOM
FLS	FIRE STATION LOCKERS AND SHOWERS
FME	FIRE STATION MECHANICAL AND ELECTRICAL ROOM
FSL	FLIGHT SIMULATOR ROOM
FSM	FIRE STATION EXTERIOR STORAGE AND MECHANICAL
FSQ	FIRE STATION QUARTERS
FT1	HANGAR CLOSED OFFICE SPACE
FT2	HANGAR WORK AREA

GAS	GENERAL ARTS & CRAFTS AREA
GRD	TOOL GRINDING AREA
HBA	MASTER BEDROOM & BATH
HBB	BEDROOM #2
HBC	BEDROOM #3
HBD	BEDROOM #4
HBE	BEDROOM #5
HCB	COAT CLOSET
HCC	STORAGE CLOSET
HCP	CARPORT
HCR	CABIN BEDROOM
HFA	FAMILY ROOM
HFB	FOYER
HGA	GARAGE
HGF	GARAGE FINISHED
HHA	HALLWAY
HJA	BATHROOM
HJB	1/2 BATH
HKA	KITCHEN - HOUSING
HLA	LIVING/DINING ROOM
HPA	PATIO/PORCH
HSA	EXTERIOR STORAGE/MECHANICAL ROOM
HSP	SCREEN PORCH
HST	STAIRWELL
HTA	HIGH TECH LAB - BUILDING SUPPORT AREA
HTB	HIGH TECH LAB - REST ROOMS
HTC	HIGH TECH LAB - SHOPS
HTD	HIGH TECH LAB - CHEM LAB
HTE	HIGH TECH LAB - ELECT/CHEM/LASER LABS
HTF	HIGH TECH LAB - CONTROL ROOM
HTG	HIGH TECH LAB - VACUUM ROOM
HTH	HIGH TECH LAB - SECURE OFFICE
HTI	HIGH TECH LAB - OPEN OFFICE
HTJ	HIGH TECH LAB - CORRIDOR
HTK	HIGH TECH LAB - CLOSED OFFICE
HUA	UTILITY ROOM
IG1	STORAGE IGLOO MECHANICAL AND ELECTRICAL
IG2	STORAGE IGLOO MUNITIONS
INF	INFANT AREA
INL	INTERIOR LOADING DOCK W/EXTERIOR CANOPY
IRR	INTERIOR ENTRY MENS/WOMENS RESTROOM
KIA	KITCHEN/WASH/STG/EMP FAC; 40-250 PERSON

KIB	KITCHEN/WASH/STG/EMP FAC; 251-650 PERSON
KIC	KITCHEN/WASH/STG/EMP FAC; 651-1000 PERSON
KID	KITCHEN/WASH/STG/EMP FAC; 1001-1500 PERSON
KIE	KITCHEN - CHILD CARE CENTER
KSM	KITCHEN (SMALL)
LA2	LAUNDRY
LAB	ELECTRICAL LAB
LAU	LAUNDRY, STORAGE, AND MAID'S ROOM
LAV	LAUNDRY, STORAGE, VENDING
LB2	LIBRARY
LIB	TAPE LIBRARY
LIT	LIVING UNIT - TYPICAL (526 SF/UNIT)
LOB	LOUNGE AND BAR
LOG	LOGISTIC SYSTEMS ROOM
LOU	LOUNGE AND GAME ROOM
LSH	LOCKERS AND SHOWERS
LU2	LUNCH ROOM 2
LUN	LUNCH ROOM
MAI	MAIL ROOM
MAT	MATERIAL CONTROL
MAU	MATERIAL CUTTING AREA
ME2	MECHANICAL AND ELECTRICAL ROOM SMALL 2
ME3	MECHANICAL AND ELECTRICAL ROOM (SMALL)
MEC	MECHANICAL
MEF	MECH. & ELECT. FAMILY CAMP
MEH	MECHANICAL MEZZANINE
MEL	MECHANICAL, ELECTRICAL, & BUILDING SUPPORT
MEM	MECHANICAL, ELECTRICAL, & BLDG SUPPORT W/MEZZANINE
MEN	CONTROL TOWER MECHANICAL & ELECTRICAL BUILDING
MES	MECHANICAL AND ELECTRICAL ROOM - SMALL
MET	MECHANICAL AND ELECTRICAL AREA
MEU	PFC MECHANICAL AND ELECTRICAL ROOM
MHA	HAZARDOUS MATERIAL STORAGE - ACETYLENE
MHB	HAZARDOUS MATERIAL STORAGE - ACID
MHC	HAZARDOUS MATERIAL STORAGE - CHEMICAL
MHF	HAZARDOUS MATERIAL STORAGE - FLAMMABLE
MHH	HAZARDOUS MATERIAL STORAGE - H.A.A.F.
MHL	HAZARDOUS MATERIAL STORAGE - LIME
MHO	HAZARDOUS MATERIAL STORAGE - OXYGEN/NITROGEN
MLG	MEDICAL LOGISTICS
MOK	MOBILITY EQUIPMENT STORAGE
MPM	MANPOWER MANAGEMENT

MSL	MISSILE LOADING AND UNLOADING
MSM	MISSILE MAINTENANCE
MSS	MISSILE SHOP AREA
MST	MISSILE TESTING
MUS	MUNITIONS STORAGE
MWA	MAINTENANCE WORK AREA
NS2	96 MAN BARRACKS COVERED WALKWAYS
NS3	96 MAN BARRACKS ELECTRICAL ROOM
NS4	96 MAN BARRACKS MECHANICAL EQUIPMENT ROOM
NS5	96 MAN BARRACKS COMMUNICATIONS ROOM
NS6	96 MAN BARRACKS QUARTERS
NUR	NURSING
OAR	OUTDOOR ADVENTURE ROOM
OAU	OPERATIONS AUDITORIUM
OBR	OUTDOOR RECREATION BREAK ROOM
OBS	OBSTETRICAL
OCO	OPERATIONS CLOSED OFFICE SPACE
OCS	OUTDOOR RECREATION CLOSED OFFICE SPACE
OFC	OFFICE - CLOSED OFFICE SPACE
OFH	OFFICE - HANGAR OFFICES
OFM	OFFICE - PMEL
OFO	OFFICE - OPEN OFFICE SPACE
OFP	OFFICE - OPEN OFFICE SPACE W/O SYSTEM FURNITURE
OFS	OFFICE SUPPLY ROOM
OFW	OFFICE - WAREHOUSE OFFICE SPACE
OGS	OPERATIONS GENERAL PURPOSE STORAGE
OIL	OIL ANALYSIS LABORATORY
OLD	OPERATIONS LOADING DOCK
OLR	OPERATIONS LUNCH ROOM
OME	OPERATIONS MECHANICAL AND ELECTRICAL ROOM
OOO	OPERATIONS OPEN OFFICE SPACE
OOS	OPEN OFFICE SPACE (SMALL)
OPR	OPERATION SYSTEM ROOM
OPS	OPERATIONS PUBLIC RESTROOMS
ORR	OUTDOOR RECREATION RESTROOMS
ORS	OUTDOOR RECREATION RETAIL SPACE
OSE	OPEN SPACE ELECTRICAL
OUR	OUTPATIENT RECORDS
OWV	OPERATIONS WEAPONS VAULT
PAP	PAPER STORAGE ROOM
PAT	PATHOLOGY
PAW	PATIENT WELFARE

PBX	PRESS BOX
PCO	PFC CLOSED OFFICE SPACE
PEN	PENETRANT/MAGNETICS AREA
PEX	MILITARY MED/PHYSICAL EXAM
PFB	PFC BUILDING SUPPORT AREA
PGS	PFC GENERAL STORAGE
PHA	PHARMACY
PHO	PHOTO/AVTR AREA
PHY	PHYSICAL THERAPY
PID	PICK-UP, DELIVERY, TRANSPORTATION, MGT, & BULK STG
PLA	PRESCHOOL AREA
PPR	PFC PUBLIC RESTROOMS
PRG	PUBLIC AREA - HANGAR
PRR	PUBLIC RESTROOMS (SMALL)
PUB	PUBLIC AREA - WAREHOUSE
PUM	PUBLIC AREA - PMEL
QEA	QUARTERS - ENLISTED/1 + 1 (506 SF/UNIT)
QEB	QUARTERS - ENLISTED/2 + 2 (544 SF/UNIT)
QOA	QUARTERS - OFFICER 01 - 02 (308 SF/UNIT)
QOB	QUARTERS - OFFICER 03 - UP (494 SF/UNIT)
RAD	RADIOLOGY
RBC	RACQUETBALL COURT
RED	READY ROOM
REI	REFRIG RECEIVING AREA
REM	RESOURCE MANAGEMENT
REP	TIRE REPAIR, TIRE STORAGE AND DYNAMOMETER ROOM
RET	GENERAL RETAIL SPACE
RRL	EXTERIOR ENTRY RESTROOM - LARGE
RRS	EXTERIOR ENTRY RESTROOM - SMALL
RSD	RECEIVING AND SMALL DRY STORAGE
RSK	RECEIVING/STORAGE/BENCHSTOCK
RSM	RETAIL SPACE (MEDIUM)
RSS	RECEIVING/STAGING AND LARGE DRY STORAGE
SAC	SATCOM EQUIPMENT ROOM
SEC	SECURITY
SEI	SECURITY CORRIDOR
SES	SECURITY SYSTEM CONTROL ROOM
SF2	STORAGE FERTILIZER 2
SFA	SERVERY; 40-250 PERSON
SFB	SERVERY; 251-650 PERSON
SFC	SERVERY; 651-1000 PERSON
SFD	SERVERY; 1001-1500 PERSON

SFS	PT / STAFF SERV / MWR
SHB	SHOP - BODY SHOP AND PAINT BOOTH
SHC	SHOP - CARPENTRY/TOOL
SHE	SHOP - ELECTRICAL/INSTRUMENT CONTROL
SHG	SHOP - GEAR BOX/AFTER BURNER/GRAPHITE SPRAY BOOTH
SHH	SHOP - HEAT & REFRIGERATION, A/C
SHL	SHOP - MACHINE SHOP AREA
SHM	SHOP - MASON SHOP
SHO	SHOP - SHOP AREA
SHP	SHOP - PAINT SHOP
SHU	SHOP - PLUMBING SHOP
SHV	SHOP - WELDING/METAL CUTTING
SHW	SHOP - WELDING/SHEET METAL
SNK	SNACK ROOM
SOR	SOUND ABSORBING ROOM
SRM	STOCKROOM
SSL	STORAGE (SMALL)
ST2	STORAGE - GENERAL
STG	STORAGE - GENERAL PURPOSE
STH	STORAGE - HIGH DENSITY
STI	STORAGE AND ISSUE
STR	SHOWER AND TOILET ROOM
STS	STORAGE SHED (3-SIDED STG FACL)
STU	STORAGE - SUPPORT SECTION
STW	STORAGE - WRM STORAGE AREA
SUI	STRIPPING PIT
SUP	SUPPORT SYSTEMS
SUR	SURGICAL
SYC	SYSTEM CONTROL ROOM (SHIELDED)
TEL	TELEPHONE EQUIPMENT ROOM (SMALL)
TRE	TRAINING & EDUCATION
UNP	UNINTERRUPTIBLE POWER SUPPLY ROOM
VCB	VAULT - CIP CONCRETE CABLE VAULT
VLA	VAULT - CLASSIFIED VAULT - TYPE A
VLB	VAULT - CLASSIFIED VAULT - TYPE B
VLW	VAULT - WEAPONS VAULT - TYPE A
VLY	VISITORS LOBBY
VMG	VEHICLE GENERAL PURPOSE REPAIR AREA - LARGE
VMM	VEHICLE MAINTENANCE AND MATERIAL CONTROL ROOM
VMS	VEHICLE GENERAL PURPOSE REPAIR AREA - SMALL
WAS	WASH BAY AREA
WDA	HANGAR WORK DOCK AREA

WDS	WOOD SHOP
WOR	WORK AREA
XAR	ADMINISTRATIVE RESTROOMS
XAS	ADMINISTRATIVE SUPPORT
XBA	BAND ROOM
XC1	CLASSROOMS
XC2	CLASSROOMS - MEDIUM GRADE
XC3	CLASSROOMS - HIGH GRADE
XC4	OUTDOOR RECREATION CLASSROOMS
XCH	CHORAL ROOM
XCI	CLINIC
XCM	CORRIDOR WITH MECHANICAL MEZZANINE
XCN	CONCESSIONS
XCO	CORRIDOR
XCT	CLASSROOM TOILETS
XCU	CUSTODIAL
XCW	COVERED WALKWAY
XDA	DINING AREA
XDR	DRESSING ROOMS
XES	EXTERIOR STORAGE
XGW	PFC GYMNASIUM
XGY	GYMNASIUM
XHE	HOME ECONOMICS LAB
XKI	KITCHEN - SCHOOLS
XLR	P.E. LOCKERS & SHOWERS
XLS	PFC LOCKERS AND SHOWERS
XM1	MEDIA ROOMS (CLOSED)
XM2	MEDIA ROOMS (CLOSED) - MEDIUM GRADE
XM3	MEDIA ROOMS (CLOSED) - HIGH GRADE
XME	MECHANICAL/ELECTRICAL
XML	MUSIC LAB
XMP	MULTI-PURPOSE ROOM
XP1	PLANNING/MATERIAL
XP2	PLANNING/MATERIAL - MEDIUM GRADE
XP3	PLANNING/MATERIAL - HIGH GRADE
XPB	PRESCHOOL BATHROOM
XPR	PUBLIC RESTROOMS
XR1	READING ROOM (OPEN)
XR2	READING ROOM (OPEN) - MEDIUM GRADE
XR3	READING ROOM (OPEN) - HIGH GRADE
XRC	RECORDING ROOM
XRE	RECORDS

XSC	SCIENCE LAB
XSL	SKILLS LAB
XST	STAGE
XTD	TEACHER'S DINING
XTL	TEACHER'S LOUNGE
XWR	WEIGHT ROOM
XWS	PFC WEIGHT ROOM
YC2	YOUTH CENTER STORAGE
YC3	YOUTH CENTER KITCHEN/BREAK ROOM
YC4	YOUTH CENTER AUDITORIUM STAGE
YC5	YOUTH CENTER MECHANICAL/ELECTRICAL
YC6	YOUTH CENTER BUILDING SUPPORT
YC7	YOUTH CENTER TOILETS
YC8	YOUTH CENTER CLASSROOMS

Appendix C: PACES BLIS/IFC GUI

Once importing starts, a status screen will display the current status of the import. The initial import consists of two phases:

1. Importing the data from the XML file.
2. Linking the data together to form a pseudo PACES project.

Once these steps have completed, the following screen (Figure C1) will appear. This tree will give you an overall view of what happened during the import. In this case, the Project was not valid and neither were the two associated facilities. (The red typeface indicates an invalid Project or Facility.)

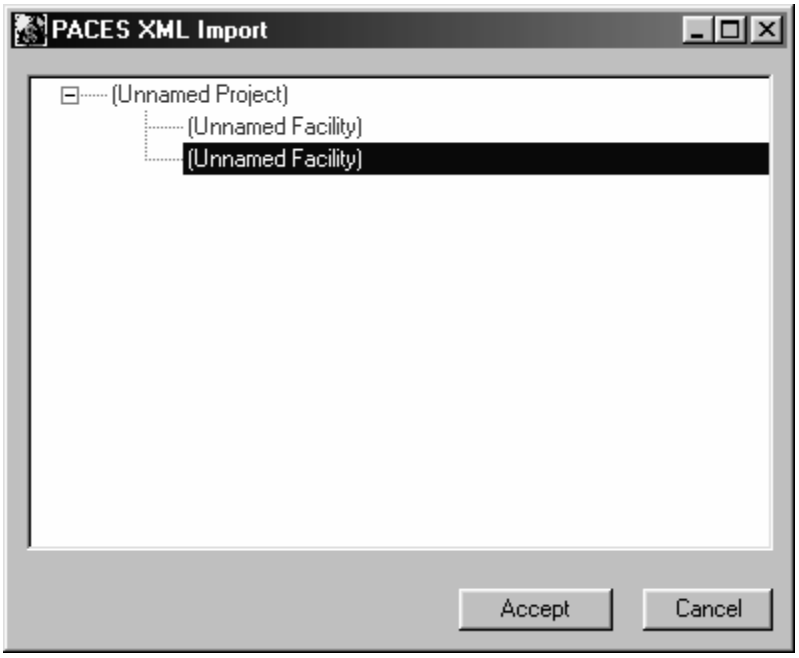


Figure C1. PACES import ifcXML dialog with errors.

When you double click the project or facility, a screen will appear to allow you to edit the Project or Facility. If you try to edit a facility when the project is invalid, a message will appear telling you to fix the project before attempting to modify a facility. The next two screen shots (Figures C2 and C3) show the project editing screen. The information from the project must be valid before you are allowed to edit the facilities or import the project.

Project editing screens

The first screen shot shows an example where the imported file contained a location and organization information and nothing else. Values that are included in the import file are used, and defaults are used for missing information. Fields with blue labels are required and must be filled in. The location cost factors and location modifiers are filled in according to the location selected. Any of these values can be adjusted.

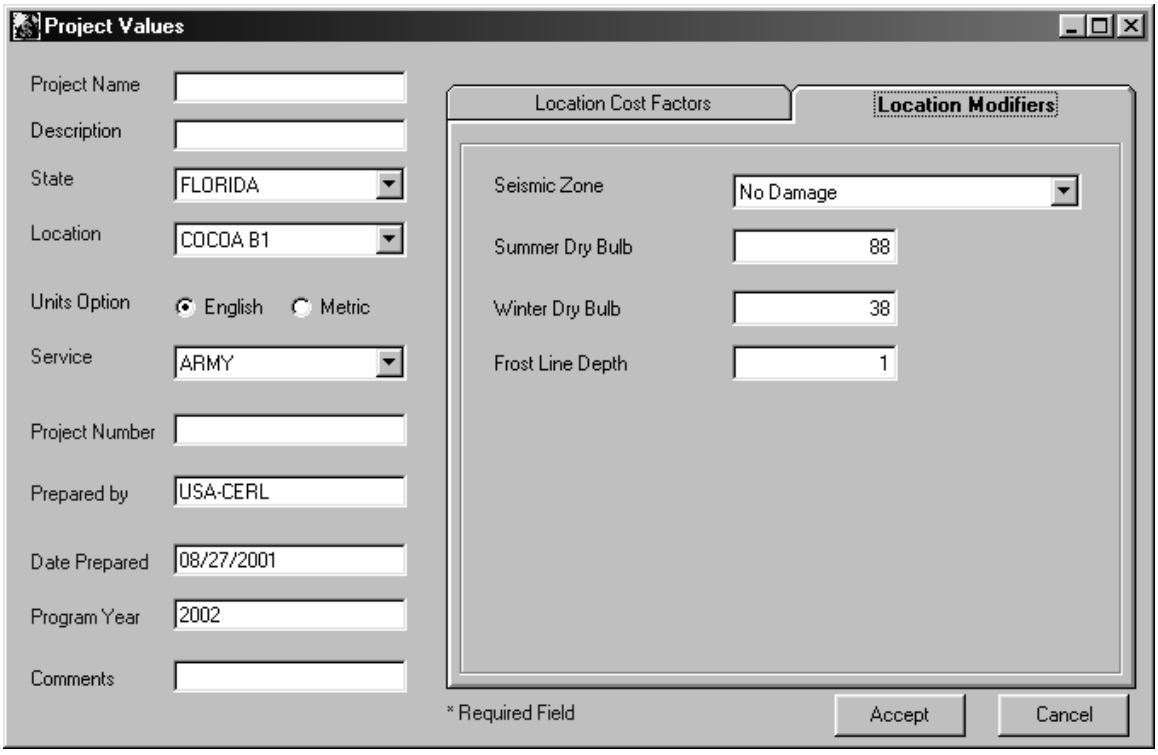


Figure C2. PACES Project Values editing screen, Location Modifiers.

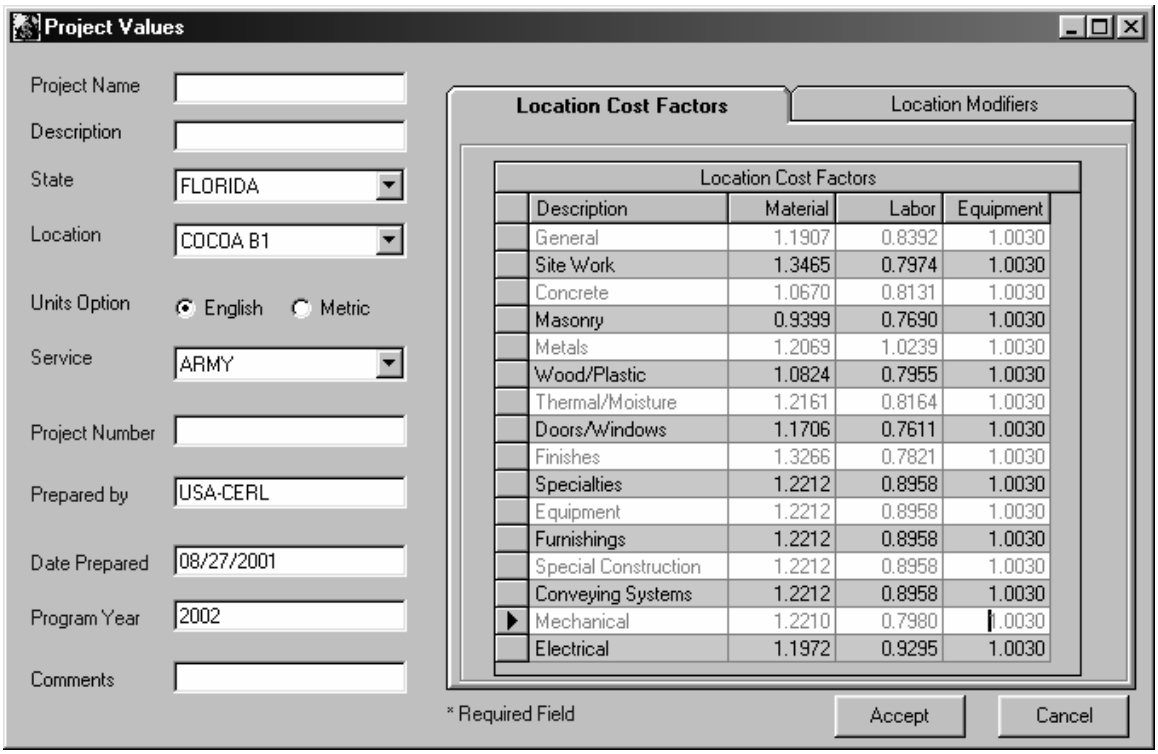


Figure C3. PACES Project Values editing screen, Location Cost Factors.

Facilities screen

As in the previous screen, all required fields are designated with a blue label. The default building and FSA information is created according to several criteria.

1. FSAs (Spaces) were defined in the XML file but no model was selected.

The FSAs are retained according to the imported file definition, and the user must select the building model to use.

2. No building model selected and no FSA's are defined.

Once a building model is selected by the user, the program will then populate the model with the model's default FSAs.

3. A building model is selected without any FSAs

The building is created using the default FSAs for the model.

4. A building model is selected, and FSAs are included.

The program will use the values imported for both the building model and FSAs.

Building size

1. Building defined with FSAs (see Figure C4).

The size of facility is based on the combined size of the FSAs

2. Building defined without FSAs, but building size specified

The specified size is used and distributed among the FSAs according to the building model selected.

3. Building defined without FSAs, and no building size specified.

The building is created just like the specified building model.

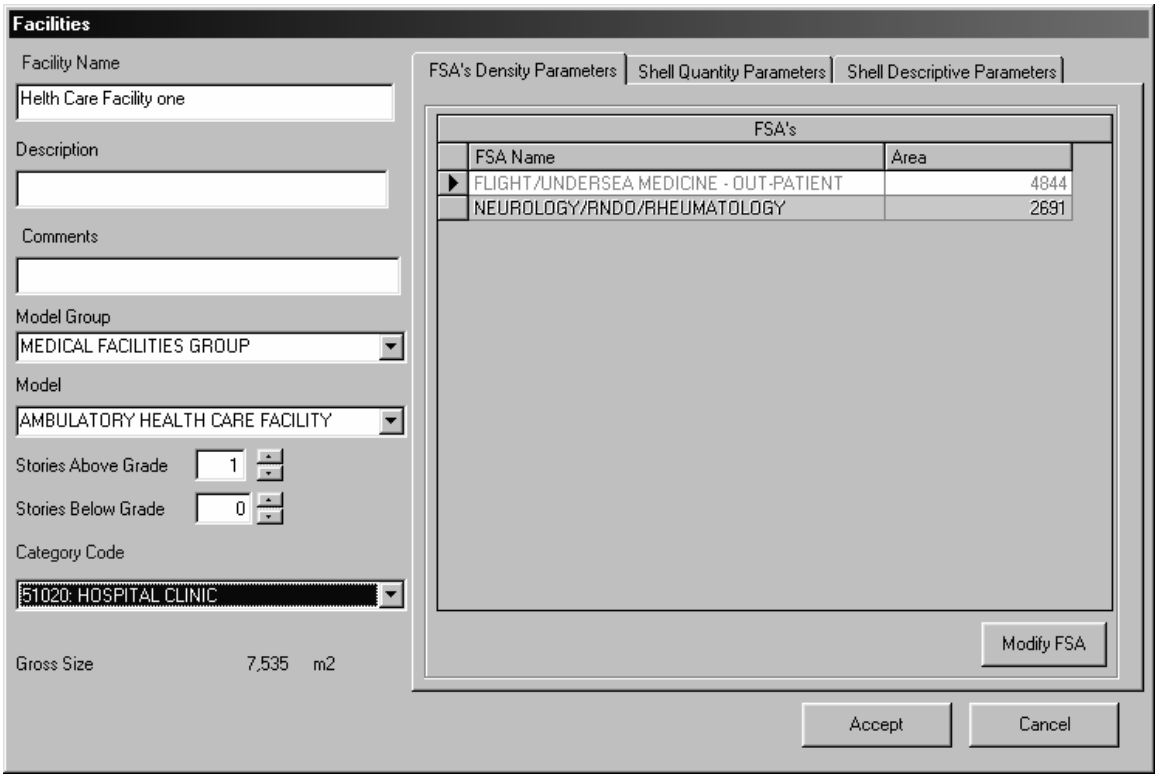


Figure C4. PACES Facilities Values editing screen, FSA's Density Parameters.

The shell Quantity parameters (Figure C5) will always show the values selected by the user. If no value was imported, the user value will be 0. The default values will show what PACES would expect the correct values to be according to its model and the size and location of the building. Any of the

user values can be edited directly in the table. The user value can be replaced with the default value by clicking on the "<" button, or use the "<<" button to replace all of the user values with the default values.

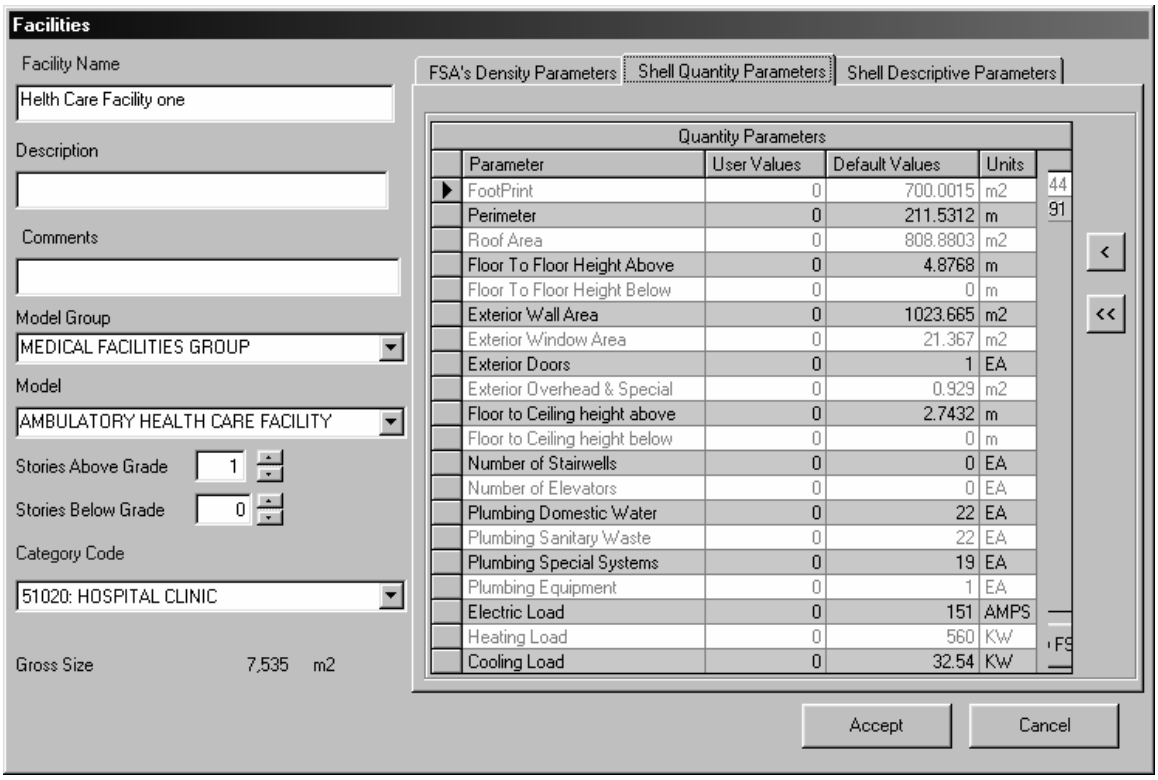


Figure C5. PACES Facilities Values editing screen, Shell Quantity Parameters.

The shell descriptive parameters (Figure C6) are populated according to the information imported from the XML file, except in the cases where an invalid value or no value is specified. In this case, the program will use the default value for the model.

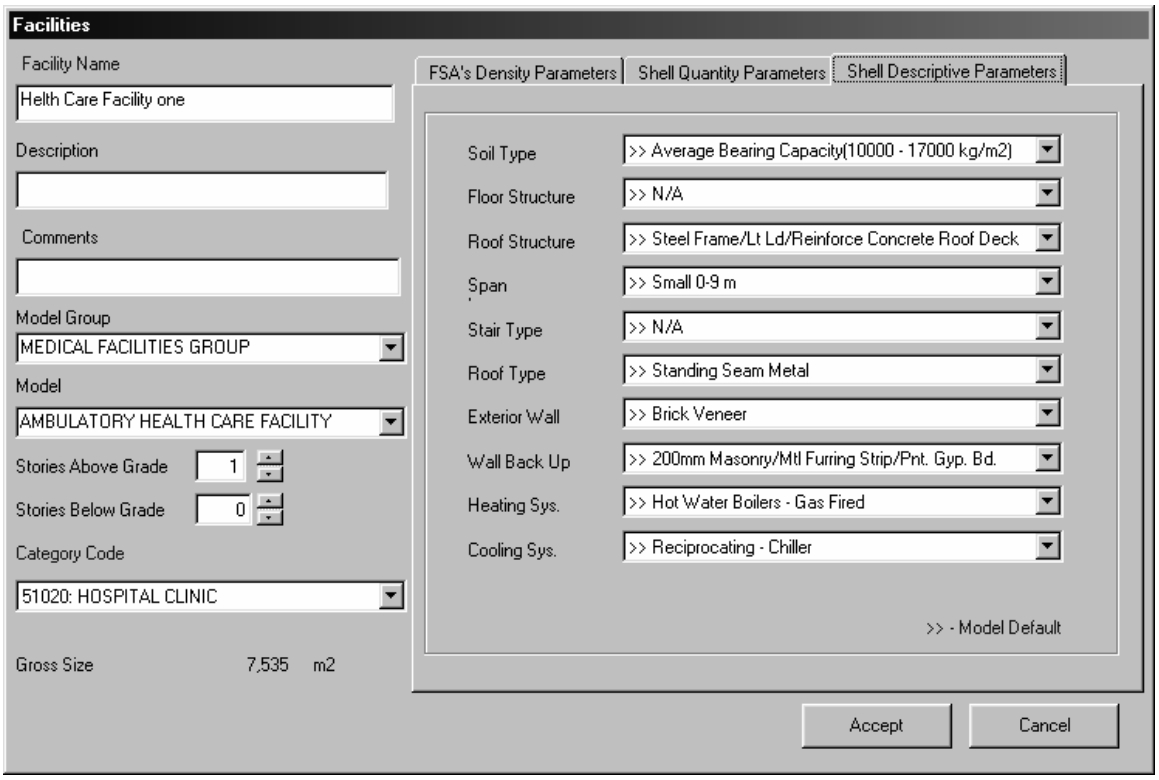


Figure C6. PACES Facilities Values editing screen, Shell Descriptive Parameters.

Once the project and facilities pass the validation, the user can click on the Accept button to import the project to PACES (Figure C7). Notice that the typeface is now black and the project and facility names are the names given to these items during the data validation process.

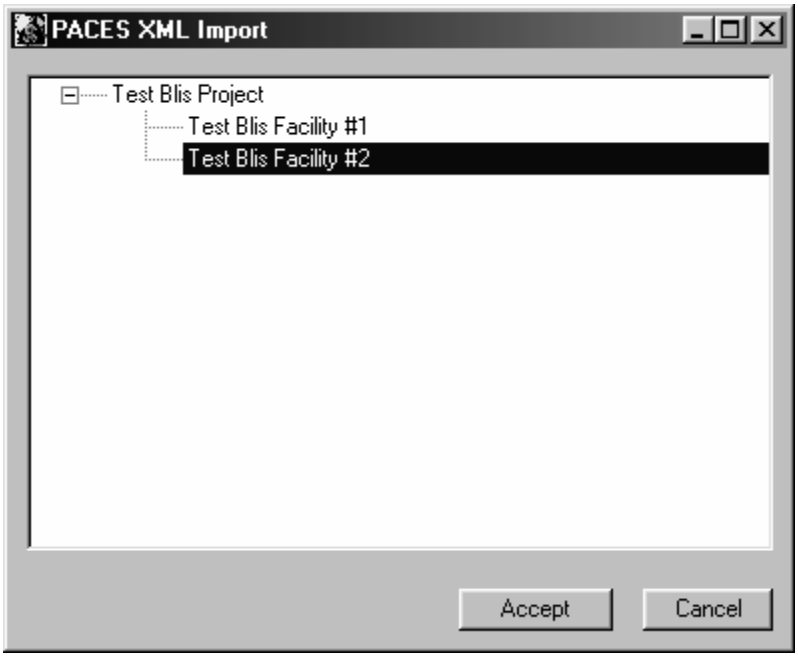


Figure C7. PACES import ifcXML dialog without errors.

Once the accept button is clicked, the project will be imported to PACES and the cost will be calculated (Figures C8 and C9). After this point, all editing of the project will be done in PACES.

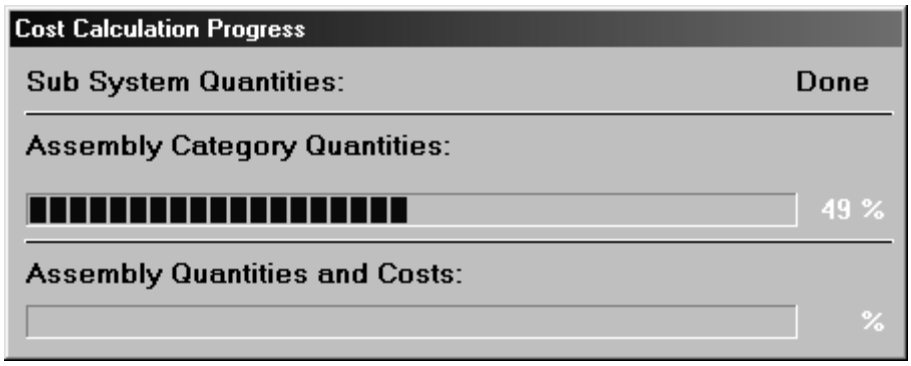


Figure C8. PACES Cost Calculation progress dialog.

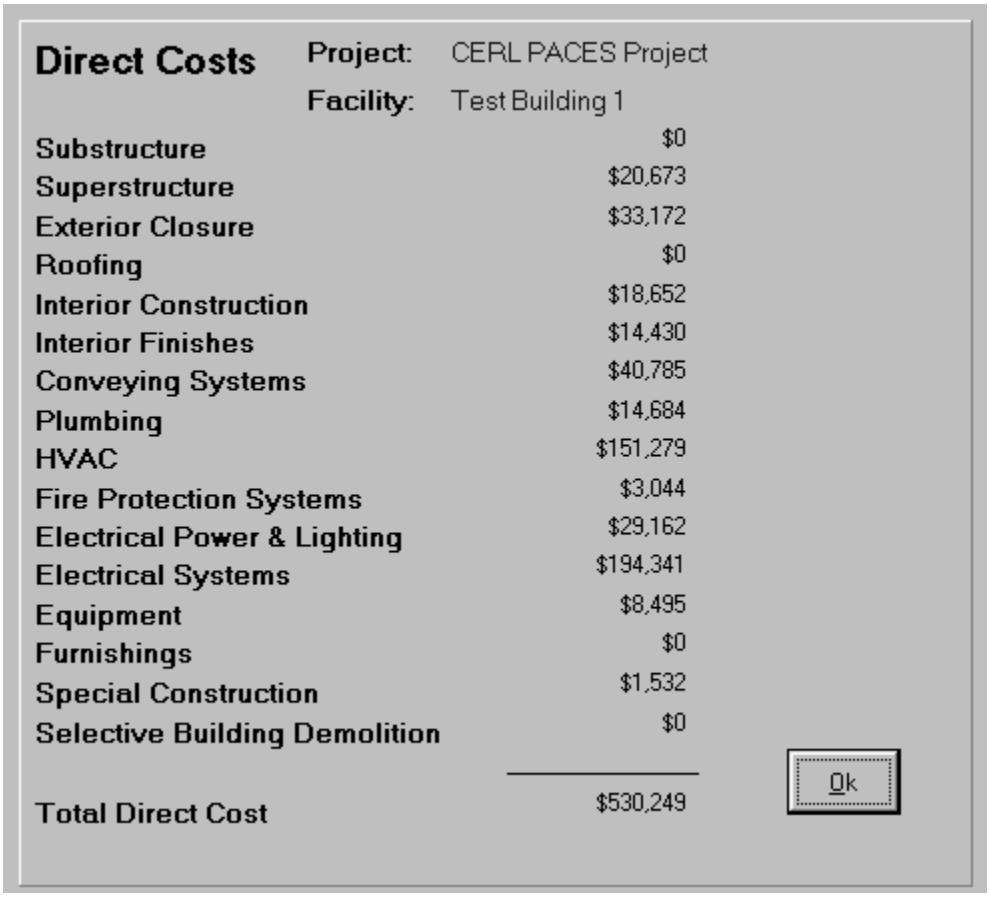


Figure C9. PACES project cost summary dialog.

Appendix D: PACES – Ifc 2x2 Criteria Mappings

PACES	Data Source	Data Destination (IFC Definition - "CC" Data)	Data Destination (IFC Definition - "LC" Data)	Questions/Comments
Project Data Fields				
ProjectName	Project.Properties Tab.Display Name	IfcProject.Name	IfcProject.Name	
ProjectDescription	Project.Properties Tab.Name	IfcProject.LongName	IfcProject.LongName	
ProjectState	Site.Properties Tab.State	IfcPostalAddress.Region	IfcPostalAddress.Region	~~This needs to be an enumeration.~~ **PACES should validate on import whether entered values are acceptable or not.**
ProjectLocation	Site.Properties Tab.City	IfcPostalAddress.Town	IfcPostalAddress.Town	~~This needs to be an enumeration.~~ **PACES should validate on import whether entered values are acceptable or not.**
ProjectService	n/a	n/a	n/a	
ProjectNumber	Project.Properties Tab.Number	IfcIdentifier.String	IfcIdentifier.String	Not so sure that this is correct. **Might want to just create the following: IfcPropertySingleValue.Name = "Project Number" IfcPropetySingleValue. nominalValue. [IfcValue. IfcIdentifier.String**
ProjectPreparedBy	Project.Properties Tab.PreparedBy	IfcOwnerHistory.OwningUser	IfcOwnerHistory.OwningUser	IfcPersonAndOrganization. ThePerson [IfcPerson.FamilyName]
ProjectPreparedDate	Project.Properties Tab.PreparedDate (jclass calender component)	IfcOwnerHistory.CreationDate	IfcOwnerHistory.CreationDate	CreationDate : IfcTimeStamp
ProjectProgramYear	Project.Properties Tab.Fiscal	IfcCalendarDate.YearComponent	IfcCalendarDate.YearComponent	
ProjectComments	Project.Properties Tab.Comments	IfcProject.Description	IfcProject.Description	
ProjectUnits	Project.Properties Tab.Units	IfcProject.UnitsInContext	IfcProject.UnitsInContext	

Facility Data Fields		PSET_FACILITY DATA		
FacilityName	Building.Properties Tab.Display Name	IfcBuilding.Name	IfcBuilding.Name	
FacilityDescription	Building.Properties Tab.Name	IfcBuilding.LongName	IfcBuilding.LongName	
FacilityComments	Building.Properties Tab.Comments	IfcBuilding.Comments	IfcBuilding.Comments	
FacilityModelGroup	Building.PACES Tab.ModelGroups	IfcPropertyEnumerated-Value.EnumerationReference [IfcPropertyEnumeration.Name (ModelGroups)]	IfcPropertyEnumerated-Value.EnumerationReference [IfcPropertyEnumera-tion.Name (ModelGroups)]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumera-tion.
FacilityModelCode	Building.PACES Tab.AvailableMode ls	IfcPropertyEnumerated-Value.EnumerationReference [IfcPropertyEnumeration.Name (AvailableModels)]	IfcPropertyEnumerated-Value.EnumerationReference [IfcPropertyEnumera-tion.Name (AvailableModels)]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumera-tion.
facilityGrossFloorArea	Building.Program Tab.Planned Build-ing.Program Tab.Actual (LC)	IfcProper-tySingleValue.Name="Gross Planned Area" IfcProper-tySingleValue.nominalValue.[If-cValue.IfcMeasureValue.IfcArea Measure]	IfcElementQuantity.Quantites [IfcQuan-tityArea.Name="Gross Area Actual"]	For LC this is the addition of all spaces area to total to the building.
facilityStoriesAbove-Grade	Number of Storey objects added in Composition view and elevation from Properties Tab of Storey object.	IfcBuildingStorey	IfcBuildingStorey	PACES import will have to determine the elevation of the IfcBuildingStorey ob-jects is greater or equal to zero and then add up to get total number above grade
facilityStoriesBelow-Grade	Number of Storey objects added in Composition view and elevation from Properties Tab of Storey object.	IfcBuildingStorey	IfcBuildingStorey	PACES will have to deter-mine the elevation of the IfcBuildingStorey objects is less than zero and then add up to get total above grade.

Facility Quantity Parameters		PSET_FACILITY QUANTITIES		
facilityFootprint	Layout Composer	Use PACES defaults	Use IfcElementQuantity and attach by IfcRelDefines-ByProperties when spaces are actually placed. If none placed, refer to CC definition.	
facilityPerimeter	Layout Composer	Use PACES defaults	Use IfcElementQuantity and attach by IfcRelDefines-ByProperties when spaces are actually placed. If none placed, refer to CC definition.	
facilityRoofArea		Use PACES defaults	Use PACES defaults	
FacilityFloorToFloor-HeightAboveGrade	Architecture Tab.BCSystem.FloorToFloorHeight (can be from any level. The lower the level, I.e. Function would override Building Level data.)	IfcProper-tySingleValue.Name="FloorToFloorHeight" IfcProper-tySingleValue.nominalValue.[If-cValue.IfcMeasureValue.IfcLengthMeasure]	Use IfcElementQuantity and attach by IfcRelDefines-ByProperties when spaces are actually placed. If none placed, refer to CC definition. This is the TotalHeight which is an IfcLengthQuantity	
FacilityFloorToFloor-HeightBelowGrade	Architecture Tab.BCSystem.FloorToFloorHeight (can be from any level. The lower the level, I.e. Function would override Building Level data.)	IfcProper-tySingleValue.Name="FloorToFloorHeight" IfcProper-tySingleValue.nominalValue.[If-cValue.IfcMeasureValue.IfcLengthMeasure]	Use IfcElementQuantity and attach by IfcRelDefines-ByProperties when spaces are actually placed. If none placed, refer to CC definition. This is the TotalHeight which is an IfcLengthQuantity	
facilityExteriorWal-lArea		Use PACES defaults	Use PACES defaults	
facilityExteriorWin-dowArea		Use PACES defaults	Use PACES defaults	
facilityExteriorDoors	Architecture Tab.Doors.Exterior Doors (From PACES Component Library)	IfcProper-tySingleValue.Name="ExteriorDoors" IfcProper-tySingleValue.nominalValue.[If-cValue.IfcSimpleeValue.IfcReal]	IfcProper-tySingleValue.Name="Exterior Doors" IfcProper-tySingleValue.nominalValue.[If-cValue.IfcSimpleeValue.IfcReal]	Will need to use the concepts of IfcConstraint and IfcConstraitUsage from IfcControlExtenion to be able to validate number specified in the planning stage and what is actually placed during detailed design.
facilitySpecialDoors		Use PACES defaults	Use PACES defaults	
FacilityFloorToCeil-ingHeightAboveGrade	Architecture Tab.BCSystem.FloorToCeilingHeight (is part of the Function Criteria)	IfcProper-tySingleValue.Name="FloorToCeilingHeight" IfcProper-tySingleValue.nominalValue.[If-cValue.IfcMeasureValue.IfcLengthMeasurel]	Use IfcElementQuantity and attach by IfcRelDefines-ByProperties when spaces are actually placed.	Is the function criteria export when the function has yet to be placed on a Storey?

FacilityFloorToCeilingHeightBelowGrade	Architecture Tab.BCSystem.FloorToCeilingHeight (is part of the Function Criteria)	IfcPropertySingleValue.Name="FloorToCeilingHeight" IfcPropertySingleValue.nominalValue.[IfcValue.IfcMeasureValue.IfcLengthMeasure]	Use IfcElementQuantity and attach by IfcRelDefinesByProperties when spaces are actually placed.	
facilityStairwells		**Use PACES defaults**	see comment field	PACES to total up number of Stair functional spaces in export file if spaces have been placed. If none use defaults.
facilityElevators		**Use PACES defaults**	see comment field	PACES to total up number of Elevator functional spaces in export file if spaces have been placed. If none use defaults
FacilityPlumbingDomesticWater		**Use PACES defaults**	**Use PACES defaults**	
FacilityPlumbingSanitaryWaste		**Use PACES defaults**	**Use PACES defaults**	
FacilityPlumbingSpecialSystems		**Use PACES defaults**	**Use PACES defaults**	
facilityPlumbing		**Use PACES defaults**	**Use PACES defaults**	
facilityElectricLoad	Electrical Tab.Loads.Estimated Loads (From PACES Component Library)	IfcPropertySingleValue.Name="Electrical Loads" IfcPropertySingleValue.nominalValue.[IfcValue.IfcMeasureValue.IfcElectricCurrentMeasure]	IfcPropertySingleValue.Name="ElectricalLoads" IfcPropertySingleValue.nominalValue.[IfcValue.IfcMeasureValue.IfcElectricCurrentMeasure]	
facilityHeatingLoad	Mechanical Tab.Loads.Estimated Heating Load (From PACES Component Library)	IfcPropertySingleValue.Name="HeatingLoad" IfcPropertySingleValue.nominalValue.[IfcValue.IfcMeasureValue.IfcElectricCurrentMeasure]	IfcPropertySingleValue.Name="HeatingLoad" IfcPropertySingleValue.nominalValue.[IfcValue.IfcMeasureValue.IfcElectricCurrentMeasure]	
facilityCoolingLoad	Mechanical Tab.Loads.Estimated Cooling Load (From PACES Component Library)	IfcPropertySingleValue.Name="CoolingLoad" IfcPropertySingleValue.nominalValue.[IfcValue.IfcMeasureValue.IfcEnergyMeasure]	IfcPropertySingleValue.Name="CoolingLoad" IfcPropertySingleValue.nominalValue.[IfcValue.IfcMeasureValue.IfcEnergyMeasure]	

Facility Descriptive Parameters		PSET_FACILITY PARAMETERS		
facilitySoilType	Civil Tab.Soil Type (from PACES Component Library)	IfcProperty-SingleValue.Name="SoilType" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="SoilType" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
facilityFloorStructure	Structural Tab.Floor Structure (from PACES Component Library)	IfcProperty-SingleValue.Name="FloorStructure" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="FloorStructure" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
facilityRoofStructure-Type	Structural Tab.Roof Structure (from PACES Component Library)	IfcProperty-SingleValue.Name="RoofStructure" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="RoofStructure" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
facilityBAY	Structural Tab.Facility Bay (from PACES Component Library)	IfcProperty-SingleValue.Name="BayType" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="BayType" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
facilityStairType	Architecture Tab.Stair Type (from PACES Component Library)	IfcProperty-SingleValue.Name="Stair Type" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="Stair Type" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
facilityRoofingType	Architecture Tab.Roofing Type (from PACES Component Library)	IfcProperty-SingleValue.Name="Roofing Type" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="Roofing Type" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
facilityWallType	Architecture Tab.Wall Type (from PACES Component Library)	IfcProperty-SingleValue.Name="Wall Type" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="Wall Type" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
facilityWallBackup	Architecture Tab.Wall Backup (from PACES Component Library)	IfcProperty-SingleValue.Name="Wall Backup" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="Wall Backup" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
FacilityHeatGenerat-ingSystem	Mechanical Tab.Heat Generating System (from PACES Component Library	IfcProperty-SingleValue.Name="Heat Generating System" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="Heat Generating System" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.
facilityCoolingSystem	Mechanical Tab.Cooling System (from PACES Component Library	IfcProperty-SingleValue.Name="Cooling System" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	IfcProperty-SingleValue.Name="Cooling System" IfcProperty-SingleValue.nominalValue.[IfcValue.IfcLabel.IfcString]	BC will have actual names. PACES import will need to compare names to their specific PropertyEnumeration.

Functional Space Area Parameters [repeated for each FSA]		PSET_FSA PARAMETERS		
FSACode	Archtitectural Function Templates Window on all levels. Under Project Program.Name	IfcSpaceProgram.Name	IfcSpace.Name if spaces are placed. Use CC convention if not.	Also depends on whether a Function Template or Function Instance is in Use. Also refer to IfcSpaceProgram.Name after placed on stories. **There will need to be validation b/t Functions in BC library and PACES specific PropertyEnumeration.**
FSAArea	Archtitectural Function Templates Window on all levels. Under Project Program.Name	IfcSpaceProgram.requiredGroupArea	IfcSpace using IfcElementQuantity when spaces are placed in drawing.	Use IfcSpaceProgram.StandardRequiredArea when refering to a Function Instance. Depending on the way the function is programmed in CC could also use the .MaxRequireArea, .MinRequiredArea
fsaDoorDensity	N/A	**Use PACES defaults**	**Use PACES defaults**	
fsaWallFinish	N/A	**Use PACES defaults**	**Use PACES defaults**	
fsaSpecialDoors	N/A	**Use PACES defaults**	**Use PACES defaults**	
fsaPartitions	N/A	**Use PACES defaults**	**Use PACES defaults**	
fsaPlumbingFixtures	N/A	**Use PACES defaults**	**Use PACES defaults**	

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
11-2007	Final	

4. TITLE AND SUBTITLE
Facility Composer™ and PACES™ Integration: Development of an XML Interface Based on Industry Foundation Classes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Susan D. Nachtigall and Beth A. Brucker

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Research and Development Center (ERDC)
Construction Engineering Research Laboratory (CERL)
PO Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
ERDC/CERL TR-07-46

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Corps of Engineers
441 G Street, NW
Washington, DC 20314-1000

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Architectural Engineering and Construction (AEC) industry has made substantial effort over the past several years to create a standard facility modeling format that better enables their different software applications to work together. This emerging standard, known as the Industry Foundation Class (IFC), is being developed by the International Alliance for Interoperability (IAI) and can be found in recent releases of commercial AEC software. With the evolution of this facility modeling standard, it is becoming possible to capture criteria and requirements during planning and design, and to reuse these data during the life cycle of the facility.

The Engineer Research and Development Center, Construction Engineering Research Laboratory is currently developing a set of facility 'architectural' programming tools, called Facility Composer™ (FC). FC supports the capture and tracking of facility criteria and requirements during planning charrettes, which are key to capturing the stakeholders' requirements in the early phases of project development. As the facility program, criteria, and requirements are chosen, these tools populate the IFC object model. This model can then be used for downstream analyses such as cost, sustainability, and physical security. As the model matures, it can be used for the facility's life cycle.

15. SUBJECT TERMS

Facility planning	Fort Future	cost estimating	PACES
MILCON	Industry Foundation Class (IFC)	Facility Composer	software

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (include area code)
Unclassified	Unclassified	Unclassified	SAR	130	